Faculty of Informatics
UNIVERSIDADE DA CORUÑA

FINAL DEGREE PROJECT
DEGREE IN COMPUTER ENGINEERING
MENTION IN COMPUTER ENGINEERING

eHand: control architecture for myo-electric prostheses

Student: Alejandro Iregui Valcárcel
Director: Adriana Dapena Janeiro
Director: Francisco Laport López

A Coruña, November 2021.

With much love, to the world.

Thanks

Always grateful for my family and friends, I want, above all, to thank my mother for the support and love you have always given me.

To all the users who trusted me to do the different tests.

To my directors as well, for believing in me and helping me with this project.

To Thais Pousada for their advice.

To the UDC-everis Chair for its financial support to purchase some of the hardware components.

Summary

eHand is a technological, social and totally Open Source project that seeks to integrate hardware and software for the analysis of muscle signals. Using sensors and electrodes, the few millivolts that are generated during muscular activity will be measured. These electrical signals will be converted from analog to digital by an Arduino. The information generated will be sent to a computer via serial port. Finally, using signal processing tools, classification patterns will be extracted on the different desired movements in order to control any type of man-machine interface. Game control and a 3D model of hand movement are contemplated.

Abstract

eHand is a technological, social and totally Open Source project that seeks to integrate hardware and software for the analysis of muscle signals. Using sensors and electrodes, the few millivolts that are generated during muscle activity will be measured. This electrical signals are converted from analog to digital by using an Arduino board. The data will be sent to a computer via serial port. Finally, using tools for signal processing, an application obtains a pattern classification of the different desired movements in order to control any type of human-machine interface with them. It currently supports control of games and a 3D model of a hand.

Keywords:

Electromyography

Microcontroller

Signal Processing

Support Vector Machine

Games

3D model

Keywords:

Electromyography

Microcontroller

Signal processing

Support Vector Machine

Games

3D Model

General index

Index of figures

Table index

Chapter 1

Introduction

The use of prostheses due to the lack of limbs in the body is a story that haunts us since before Christ. Due to the limitations of the time, fixed implementations were developed whose usefulness was very limited or none at all. Over the years and with the help of advances in electronics and mechanics, humans have been able to improve them. Currently, there are advanced prostheses that are capable of giving the person back a significant variety of movements. Even so, due to factors such as the production market or technology, we find products that are not available to everyone. One of the factors that limit its use is the price/utility ratio.

1.1. Motivations

The main motivation of this project is to advance the development of a low-cost prosthesis. The current price of a commercial prosthesis is around €60,000, something that is unaffordable for many people and, above all, in cases of very young patients who need several prosthesis changes throughout their growth. This is a medium-long term objective that exceeds the completion of this Final Degree Project (TFG), but one that we are sure we will achieve in the future. This work addresses some of the steps to follow. The path to achieving the first achievements has not been direct. During the study and its execution, problems arose that had to be resolved and which we will comment on in this report.

Regarding the ease that exists to access development tools in this type of work in particular, whether sensorization hardware elements, Analog Digital conversion and specialized software, it was difficult to find a platform with which to work easily and at a low price. . For this reason, in this TFG a study of the elements is carried out with the aim of helping to identify the best options and offer Open Source instruments that can be used by anyone interested in the subject. However, The development of tools is of little use if there is no easy access to documentation or dissemination on the subject. From an engineering student's perspective, it was difficult to find specific information on Electromyography (EMG) that was easy to digest. That is why another motivation of this TFG is to offer the necessary details to understand how EMG signals can be transformed into orders for the movement of a prosthesis or any other machine. That is, we intend to explain all the steps to follow and help disseminate them.

In order to ensure that electromechanical prostheses are affordable and viable to become a product, it is also interesting to analyze other uses that could be given, and are being given, to this technology. The more it is used, the larger the market will be created, the costs will be lower, there will be more developments, the better the competition will be and, ultimately, the quality and accessibility of the products will increase. This has led us to not focus the development only on the prosthesis, but we have also studied how the final product of this TFG can be used for rehabilitation and even for entertainment.

In a system like this, which will be integrated into a person's body, there will be certain physical requirements such as implantation size and weight that will be an obstacle for many hardware architectures. It will then be necessary to study them to meet the needs of patients and specifically, in digital signal processing. It is currently very common to see specific solutions or custom designs that meet requirements for low consumption, high processing capacity and even connectivity.

With respect to the latter and motivated towards the study of the hardware and software co-design of the system, it is impossible not to see the current rise of machine learning as a tool for signal processing and how nice it is to discover that it is a completely natural and useful for classifying movements using muscle signals.

Finally, there is the personal will of the student to create an Open Source software community for the study of this type of signals, with the intention of providing what is necessary to allow any type of work related to EMG, always under the motto of " sell hardware, not software."

1.2. Myo-electric prostheses

Currently, the most used and simple to design prostheses are hook type (they only open and close the hand). During this section we want to expose the general structure for its control using EMG signals. Its complete development exceeds the objectives of this TFG, but knowing its needs will help to understand the parts addressed in our development.

The hardware of a prosthesis must be capable of performing the following:

- Read EMG signals.
- Analog-Digital Conversion.
- Process these signals in the form of orders.
- Monitor the status of mechanical elements such as motors and respond with action.
tion.
- Have a good firmware to verify the status of mechanical components, battery, updates, etc.

Figure 1.1: Open Bionics children's prosthesis, Hero Arm [1].

First of all, it is very important to determine the set of components that are going to be used for each of these needs. In a perfect world we should be able to use the same configuration for any prosthesis, but, in reality, it will depend directly on the type of amputation or the patient. For example, if we want to design a prosthetic hand for a child like the one shown in Figure 1.1, we will need to meet very specific size and weight requirements. Achieving this without sacrificing certain aspects such as autonomy, the number of components or computing power is a remarkable job. Therefore, it would be interesting to find a balance that would allow different prostheses to be produced without the need to reinvent oneself at every moment.

Software is also an important part for a system like this. It is clear that for the correct design we would depend on the configuration of components chosen. But, if we worked on a series of tools that would speed up the production process, we would be taking a big step. We therefore believe that there is a need to produce free software, which would save hours of work and testing. The collaboration of a community would push developments to achieve a scalability that is not achieved through individual solutions. We believe that these are currently the biggest problems facing a system of this type. There is no sufficiently general hardware and software solution that satisfies these requirements and that developers can easily work with, but rather each organization provides a solution that is too specific and born from scratch, which implies a very high development time and cost.

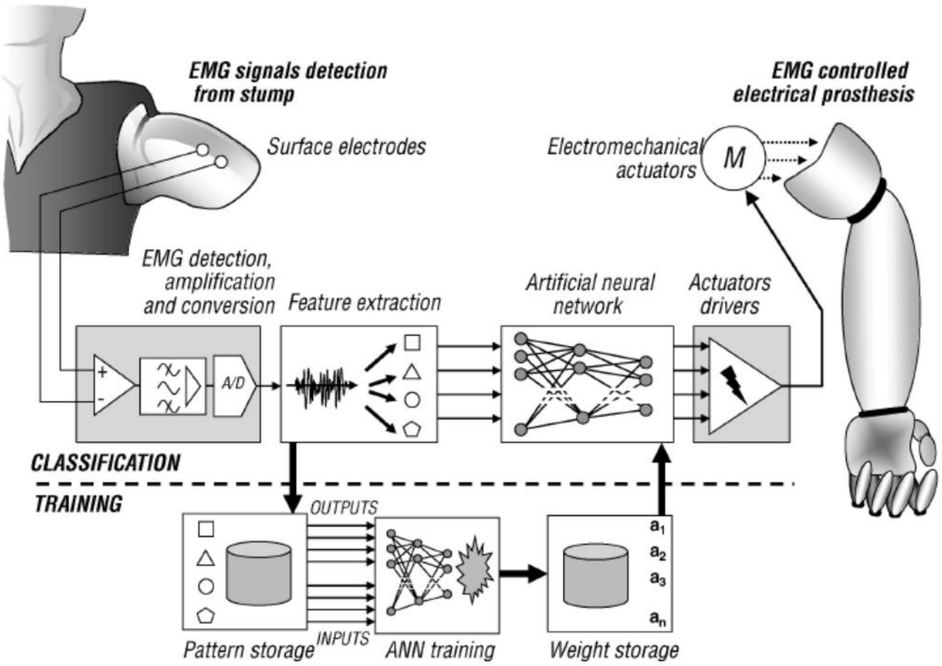


Figure 1.2: Schematic of a myo-electric prosthesis [2].

By doing an analysis exercise on Figure 1.2, at each stage certain requirements could be defined that could provide more general solutions applicable in different cases.

In the first stage of signal acquisition, it is normal that the quality of the circuitry varies by each manufacturer, but a certain standard of quality and characteristics could be defined in the analog signal they generate. For this reason, in chapter 2 we will review several sensors.

We will see during the review that these sensors share characteristics in the analog logic signal they produce due to the compatibility they seek to have on Arduino-type development boards. That is, this type of microcontroller in one way or another is forcing manufacturers to follow the same trend in terms of the sensory products they offer.

With respect to the number of sensors, a balance could be defined between the type of amputation and the number of movements to be recognized. The leg, for example, has fewer movements than a hand, so it would not be logical to use the same number of sensors in all cases.

you are Likewise, this analysis will also directly depend on whether the following phases of feature extraction and classification are able to take advantage of them.

If we have a certain standardization in signal acquisition, the following 2 stages could be developed in a generalized and Open Source way. In the extensive bibliography consulted, each author presents different solutions and configurations. If every researcher is going to reinvent the wheel all the time, many of these developments will probably remain unused or forgotten. Community work could be done on a software design dedicated to solving the problem of extracting the information necessary to mimic the human limb based on sensorization conditions.

It is also important that in the architecture there is a division of hardware, signal processing and the rest of the components. That is, what gives the system the ability to perceive the real world and act intelligently should be separated from the more sequential behavior that we commonly know. What we are looking for is to want to give flexibility and generality to the system so that it is capable of easily hosting any of the software processing methods that have been developed for this cause.

Once the most important phases in the production of a prosthesis are understood, let the manufacturers dedicate their efforts to the lowest level production and sale. The interesting thing is that each seller defines the quality/price relationship they want to give to their product, to offer different alternatives to the market.

1.3. Goals

The final goal, in the medium-long term, is to have an arm-hand prosthesis like the one described in the previous section. However, this requires technological development that is outside of this TFG. For this reason, six objectives were set in this project.
concrete:

- Make a review of the most important information and current state of technology used in EMG.
- Find and analyze signal acquisition hardware that is low-cost and useful for experiments.
- Build free software type software tools for the treatment and classification of signals.
- Use these tools for different applications like a game and controlling a 3D model.
- Do experiments and test the applications on real users.

- Document the entire process and highlight the most relevant aspects that are learned through the experience of working with the equipment so that, in the future, anyone can easily replicate what was done in this TFG.

1.4. memory organization

This document aims to reflect the student's experience in his journey to discover from scratch what it means to build a system of this type. The memory is organized in the form of a guide, where we will begin by introducing basic concepts: the biology that exists behind these signals, the instrumentation for their acquisition, the processing, and finally, the chapters where we will talk about the developed applications and the experiments with users.

- Chapter 2: State of the art. The current situation of EMG technology will be discussed and some of the biological concepts on which it is based will be introduced.
- Chapter 3: Developed architecture. The theoretical procedures and concepts on the tools used to process the signals used will be described and the final applications will be discussed.
- Chapter 4: User testing. The experiments carried out will be presented and described.
- Chapter 5: Methodology. The methodology used, planning and development costs are described.
- Chapter 6: Conclusions of the work and future directions.

Episode 2

State of the Art

In this chapter the current situation of EMG technology will be discussed both at the level of hardware that can be used as well as applications that have been made by others researchers.

All analysis, whatever the development, is based mainly on the concept that with electromyography, we seek to extract information from muscle activity to recognize movement patterns and extract characteristics of each of the muscles involved.
That is why we will begin by explaining some of the biological concepts on which EMG technology is based.

2.1. Basic anatomy

Muscles are a type of soft tissue that can contract through nervous impulses, generating movement and allowing mechanical work. Electromyography is a technique that measures the electrical activity generated by the passage of the nerve impulse, which causes depolarization in the muscle cell membrane during excitation. If the depolarization reaches a certain threshold value, an action potential is generated [2, 3].

Therefore, EMG indirectly measures muscle activity, since it is capable of determining whether the nervous system is actively recruiting muscle fibers.
during a task [3]. During contraction, an electrical potential is produced that is measurable in the muscle membrane, as shown in Figure 2.1. This phenomenon arises in cellular units called motoneurons [2], which in turn are organized into motor units (which come to establish contact with an average of 150 muscle fibers). The interesting thing about this is to understand that electromyographic signals are composed of each of these potentials and, therefore, it is possible to decompose the signals to obtain the activity existing in each of these cellular organizations.

Applied to prosthesis control, it is important to recognize and study the areas where

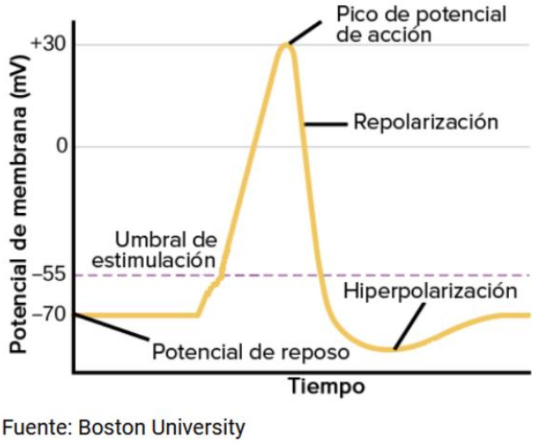


Figure 2.1: Action potential in the muscle membrane [3].

They are going to do the readings. This will depend on the type of amputation, the number of signals that we want to use for control and even characteristics such as the percentage of fat that the patient has in their body and age, among others.

Although the ideal prostheses are those that would allow all hand movements to be performed, in practice the most used movements are those of the "pincer" type that consist of opening and closing the hand [11]. More specifically, as we will show later in this report, we have focused on the development of a prosthesis with 2 sensors, for which we will measure the activity of the flexor carpi radialis and the extensor carpi radialis longus. These muscles located in the forearm are involved in various hand movements. The flexor carpi has the action of the main flexor of the wrist, with a tendency to abduction and pronation. The extensor carpi radialis longus performs a slight flexion in the elbow joint and an extension in the wrist joint, also collaborating in closing the fist and radial deviation. With this placement of the sensors, we will seek to extract characteristics of the flexion and extension of the wrist and, if possible, also the closure of the fist [12].

In this area of the forearm we could differentiate the actions even coming from each finger, this would imply greater and better sensorization and a superior intelligence capacity in the controller to distinguish the signals coming from each movement. This would be the most interesting if we want to imitate the behavior of a hand as much as possible. However, we would encounter certain restrictions, such as increased cost, a longer patient adaptation and training period, and even an increase in the size or weight of the implantation.

2.2. Hardware Review

The choice of hardware is key to creating an EMG-based system and must take into account both the part of guaranteeing the correct acquisition of the signals and that related to the characteristics of the users who are going to use the prosthesis. Below, we present the most notable aspects of the study that has been carried out in this TFG to decide the hardware to use in the development.

2.2.1. Electrodes

The electrodes are like small microphones that are used to listen to the muscles. There is a wide variety of these on the market, so it is necessary to first understand the use given to each type to obtain the best quality of surface EMG signals [12].

As for shapes and size, the choice will depend on the muscle area or muscle that we want to monitor and the distance between electrodes that we apply. The most common are between 0.5 cm and 1 cm in diameter and a distance of 1 cm and 3 cm between them. For example, if we wanted to measure activity in a specific muscle on a person's face, it would be best to use the smallest possible diameter, precise placement, and 1 cm apart. On the other hand, if what we are looking for is to extract the activity of an area such as the back in which there are several muscles, it would be necessary to use electrodes with the largest contact surface and positioned at a greater distance between them [12].

Regarding its composition, the most common is to use electrodes made of a plastic impregnated with silver that will be coated in a thin layer of silver chloride to help stabilize the electrical potentials in the skin. You can differentiate between 2 types: dry and wet electrodes. The difference will lie in the use of a conductive gel between the electrode and the skin, which helps to conduct the electrical charge better. In general, the most comfortable to use are the dry ones, since they allow easy interaction and can be repositioned without much problem, which will be very useful because their positioning is crucial (and we emphasize the latter because it has caused a lot of pain head).

Regarding the format in which they usually come, we think that it is time to reinvent them a little. To be able to read several muscles in the same area, their placement can be quite
cumbersome

For measurements, two emitters are commonly used (one at each end of the muscle to be measured) and mass (normally located in the bone area) which allows a cleaner signal to be extracted. As an example, in Figure 2.2 you can see the placement of four wet electrodes in the area of the arm corresponding to the wrist flexor and extensor muscles (which we talked about in the previous section). The figure above shows the electrodes placed on the flexor, while in the figure below they are placed on

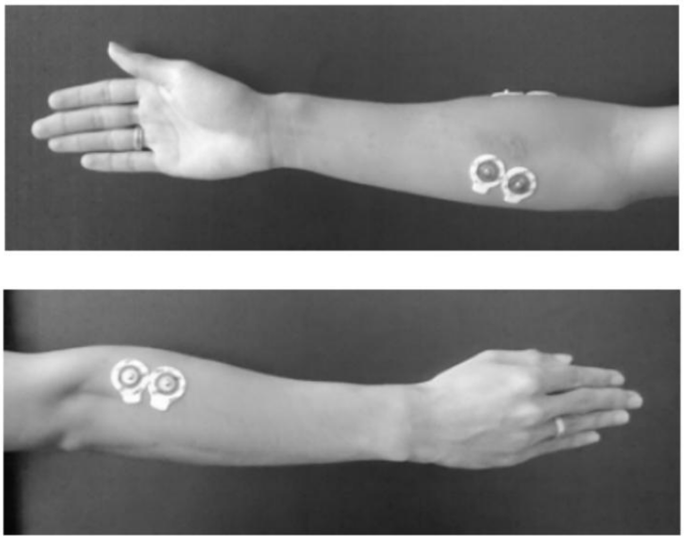


Figure 2.2: Electrode placement during experiments [4].

the extender.

Before starting development, different types of electrodes were analyzed. Table 2.1 collects the most important characteristics. It was possible to verify that both electrodes work correctly when integrated with the EMG plates. The brand passive electrode Olimex has a larger size and cost, but the fact that it is dry allows for placement faster and cleaner. Since they will only be placed in 2 areas, the final price is more than reasonable for this project. Based on this study, we have decided to use the brand Olimex.

Model	Maker	Type	Dimension	Price
Olimex passive ECG and EMG electrodes		Dry	1cm	€10
Electrodes with disposable patches for ECG and EMG	A stranger (China)	Wet	0.5cm	€5

Table 2.1: Electrodes used in this project.

2.2.2. EMG plates

What would then be the easy way to read these signals? To date, the first resource of The information you turn to to answer something like this is the Internet. By investigating, you can find that there are some American manufacturers of printed circuit boards (PCB) for development that offer solutions that are difficult to access and at high cost, which does not adapt to the objective set in this TFG. As an example, details can be found on the Ossur [13] and Biosignal R&D [14] websites.

There is a YouTube channel called "Biomakers" [15] with very good tutorials on all types of biological signals (muscle, brain, etc.) where they even teach the designs for the circuitry necessary for measurement. Therefore, we will begin this section by talking broadly about their composition, since it is surprising how simple they are.

The basic principle of a sensor is to map a real quantity such as light or temperature into mathematical functions. We extract values from the world around us in order to be able to represent them in a transducible way. In the case of the EMG, since we are reading the few millivolts that are generated during muscular activity, there will be no need to make transformations between magnitudes between our signal and the sensor, since it will simply serve as an amplifier of the activity to obtain a manipulable signal.

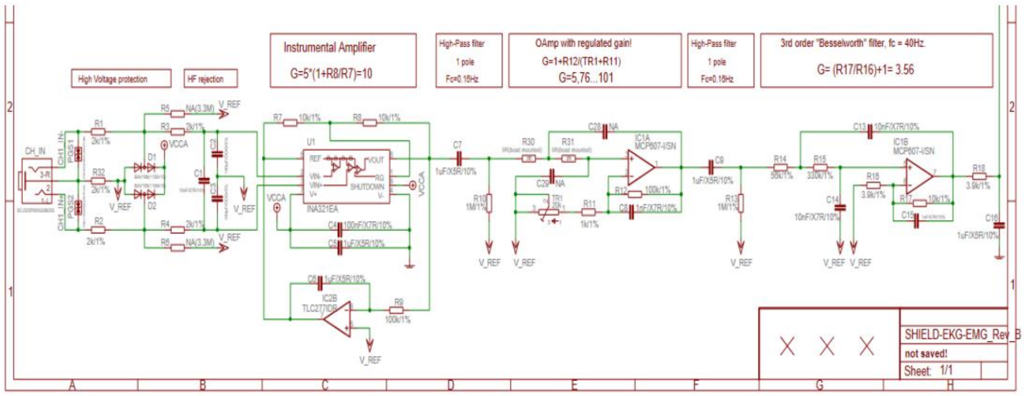


Figure 2.3: Schematic of the Olimex board [5].

The circuit in Figure 2.3 refers to the SHIELD-EKG-EMG board from Olimex [5]. As its name indicates, it is a "shield" that can be stacked on Arduino Uno type boards. It has all the components to be calibrated and connected to it and allows the reading of several channels. This part of its schematic corresponds to that of the electromyographic sensor.

Starting from the far left in the schematic, we see that there is a "CH_IN" connection where we have 3 connection terminals: L, R and Ground, each corresponding to the terminals on the electrodes. Broadly speaking, what we can see is that the currents collected by these elements will go through several phases of amplification and filtering. Starting with protection components, we move to a stage where we reject high frequencies and then amplify the signal using an amplifier operational of a certain profit. We then continue with filtering stages for the characteristic 0 Hz frequency in the direct current to, finally, reach a stage that I believe deserves some explanation.

In this case it is a phase where we will use a regulated gain amplifier. And why regulated? Well, during the biological concepts section it was explained that this type of sensors will be affected by certain physical characteristics of the person and, it may happen, that they do not have sufficient sensitivity to collect signals from the surface (either due to the fat that separates the sensor and the muscle or also due to the low muscle mass in the area). For this reason, the Olimex PCB allows us to adapt to any situation.
However, the calibration for this board, in our opinion, is a bit cumbersome, since it is done physically and using a pulse width signal (PWM) that it generates, so it would be interesting if it could be done physically. completely automatic, through
software.

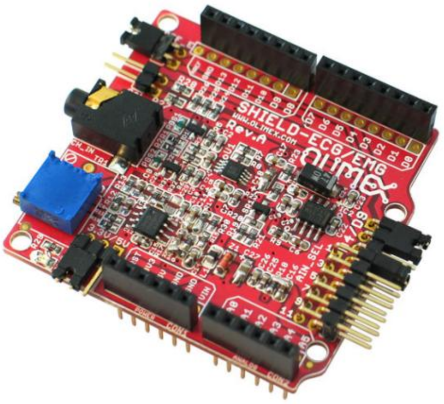


Figure 2.4: Olimex EMG sensor [5].

Figure 2.4 shows the plate. The raw signal it provides does not allow easy interpretation of what is happening and presents noise in its information. It is a completely positive wave and the values during a contraction or rest are almost indistinguishable. In this situation it would be practically impossible to extract features to classify movements. The existing documentation does not help to understand the correct use of the plate to read EMG, nor does it explain the possible origin of the noise.

However, we found a C++ library on the Internet to filter EMG signals that was tested in this TFG with the Olimex sensor. The filter described is a band-pass filter between 20 and 150 Hz, resulting in a wave with a positive and negative component that suggested the idea of applying a complete rectification to extract the power of the signal (theoretical concept that will be explained later). and with which good final results were achieved.

The second plate analyzed is of Chinese origin and its manufacturer is unknown, since it is

They buy from a supplier that does not offer information. According to what can be read on the Internet, it seems that it could be a case of "copying" in which they base their designs on models from other manufacturers. It can be found on Aliexpress for a price of €15, it comes in a pack with the electrodes, patches and sensor. Figure 2.5 shows a photo of the plate and, as can be seen, it has a very adequate size for comfortable development with ease of placement and transportation.

This board needs a power supply of +9 V and ÿ9 V, it uses the positive and negative reference to carry out operations, the output is given through the "SIG" terminal and it is an analog signal between 0 V and 3 V , that is, completely positive (this makes sense because the Arduino is unable to read negative voltages on its analog terminals).

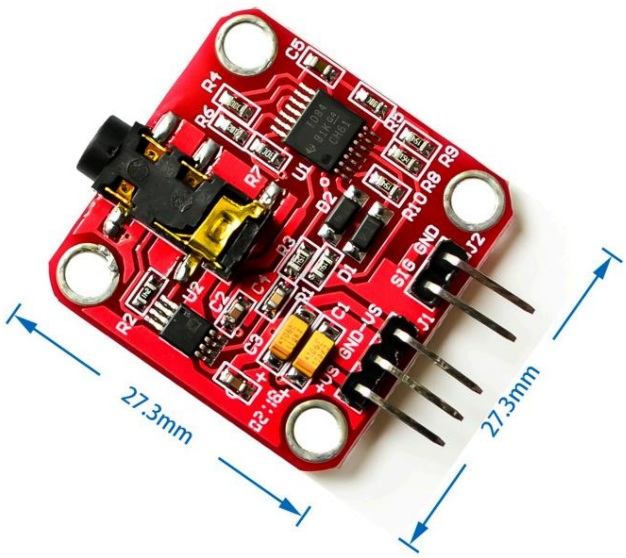


Figure 2.5: EMG sensor of Chinese origin [6].

The assembly can be seen in Figure 2.6 and is quite trivial, it can be done by anyone at home with an electronics starter kit. You can see that with 10 cables everything would be assembled, so it would be very easy with this size and even a Nano-type Arduino, to design a casing on a 3D printer to attach everything comfortably.

After having placed the electrodes, it is easy to start reading the first values related to muscle activity. To do this, we configure the serial port in the Arduino code and, within the loop, we read the analog inputs to which the electrodes are connected. The implementation of said program is simple, so it will not be detailed in this report. However, said source code is available in the

GitHub repository [16].

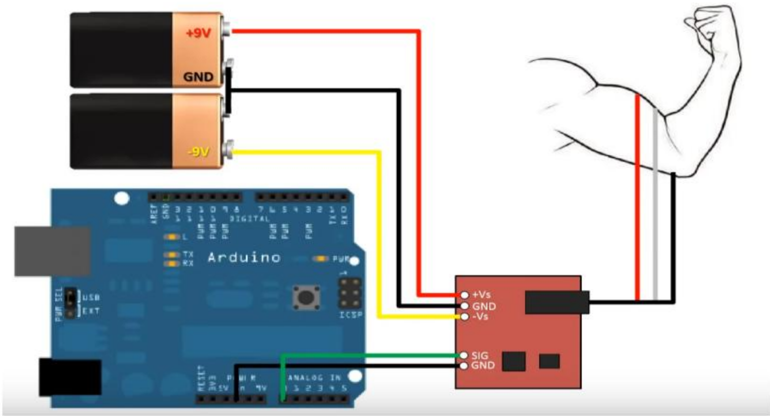


Figure 2.6: Assembly diagram for the Chinese board [6].

The tests carried out and the developments presented in this project work differently with the Olimex board and the one of Chinese origin: the 2 sensors return the same result and essentially what differentiates them is their physical format. The implementation of the system in this work is abstract enough for modularity to exist between the hardware and software layers. The only thing to highlight is that the Olimex board would only work in a scenario in which it can be stacked on an Arduino type Uno. On the other hand, the board purchased on Aliexpress has a price that is very affordable, the size and usefulness offer Cida is very good, in addition, it is easy to put to work and also allows integration with any other socket with analog input in its circuitry. As we can see, each PCB has its advantages and disadvantages.

There are also 2 American plates that were considered for the development of this work. The Myoware sensor [7] from Advancer Technologies and Gravity [8] from OYMotion.

In Figure 2.7 we have the Myoware board. As we see on the manufacturer's website, it works with a single supply of between +3.3 V and +5.0 V. It has the ability to choose between an analog output that would correspond to the completely rectified or raw muscle signal (concepts that we explain during chapter 3). You can also physically adjust the equipment's gain. Another quality of it is the format in which it is presented, it has the electrodes welded in its casing. Its design would allow us to use it in any scenario, coupled to any type of microcontroller that is capable of powering it and reading its analog signal. However, this sensor was discarded for now due to the extra costs of importing it.

Figure 2.7: 1-channel Myoware sensor [7].

The other PCB, the Gravity sensor, can be seen in Figure 2.8. It works with the same power supply that is used for the previous one and could also be coupled with any socket that is capable of supplying power and reading analog signals. Although, in this case, the Gravity plate is only able to read the rectified muscle signal. Its design consists of 2 parts: the PCB with the sensory circuitry and the dry electrodes. It was also discarded due to the extra costs generated during its importation.

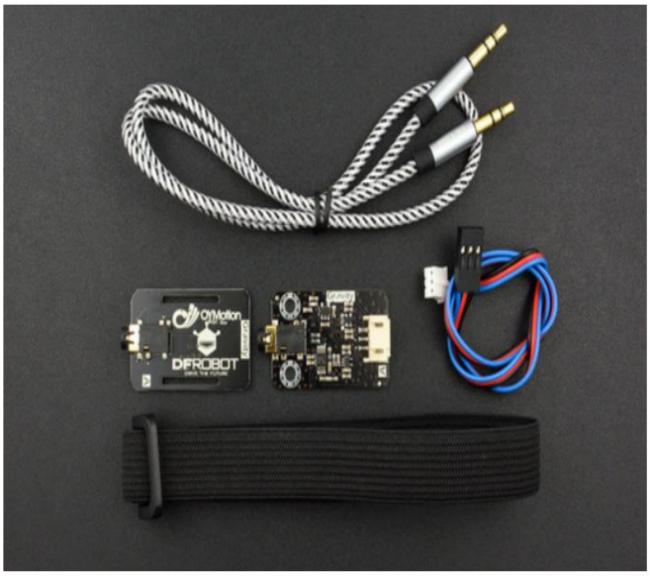


Figure 2.8: 1-channel Gravity Sensor [8].

Finally, an attempt was made to use the PCB shown in Figure 2.9. This board has a sensor that could also be used for this purpose. This is the Open BCI Cyton board [9] and is currently being used in university projects related to electroencephalography (EEG). It has software support, a program and libraries to read and analyze biological signals. It also has several input channels in which the electrodes would be placed. However, these connections are not compatible with the electrodes used during this project, because of this, the plate was discarded for now in the experiments of this work.

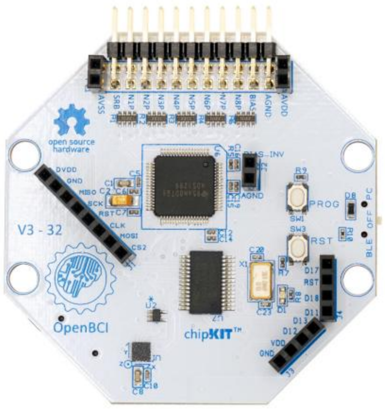


Figure 2.9: 8-channel sensor, Cyton from Open BCI [9].

23. Known developments

During the introduction it was commented that it was necessary to do an analysis of the current market and uses of this technology. It is a topic that could be discussed for chapters, but that is not what we intend. This document is intended to serve as a simple guide to follow, that is, to allow the reader to delve deeper into the concepts discussed in this report.
It will then be more interesting to summarize the current panorama of this technology where we will highlight developments that have had great influence on this project.

The Spanish company MDURANCE [10] supplies this type of equipment to both sports and occupational therapy clinics. It has achieved a good reception among its clients, offering both hardware and software ready to use by specialists in their sessions.

It is undeniable that this type of equipment is not something new and that it has been used for a long time in the area of sports and research. Likewise, it must be said that this is material with a cost that few places and individuals can afford, so it is common rethink its true usefulness. Another reason, no less important, is that it is useless to have equipment for EMG readings if we do not then have good software support that facilitates interaction with these signals. In the end, what is interesting is the information that can be extracted after the different processing (a matter that is no longer available to everyone).

This is what MDURANCE provides, a product from start to finish, buy and wear.

As can be seen in Figure 2.10 and Figure 2.11, the equipment is used to obtain readings of muscle activity during certain movements and thus offer the specialist more detailed and individual training on muscle synergies, decompensations and fatigue, among others.

The most interesting thing about the product offered here is the scalability to implement a system that is not only based on electromyography and grows towards a clinical assistant with the ability to draw conclusions based on different data collected during the sessions.

Right now, they also have Videofeedback and progress monitoring.

The current problem that we see in their business model is in the acquisition of their product. It is necessary to first contact them and this suggests that they work on demand and the prices can surely scare more than one person. After having some conversations with them, possibly this situation develops mainly due to hardware supplied from third parties, if we add to this the few companies in this equipment niche and therefore the low competition, the costs will surely be high.

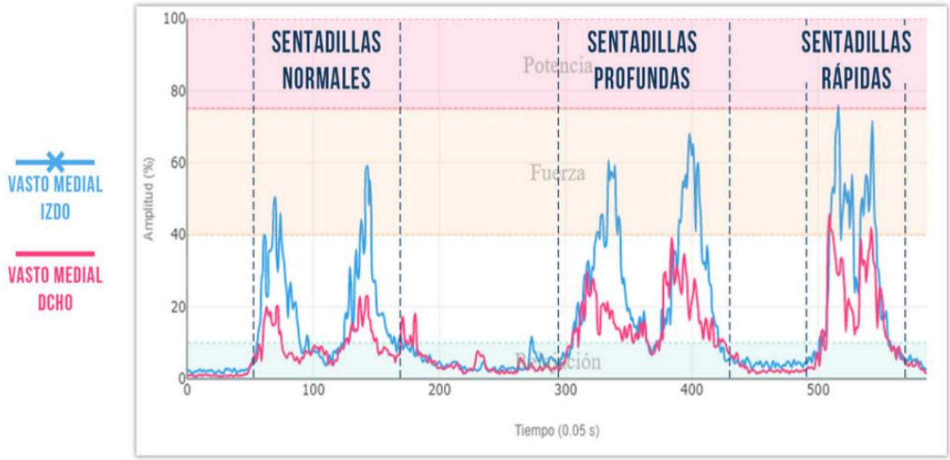


Figure 2.10: EMG during the squat [10].

Finally, comment that the electromyograph from this manufacturer has 4 channels and with Bluetooth technology, it has an autonomy of 12 hours and a sampling frequency of 1024Hz. Are characteristics could in principle be achieved with our plate of Chinese origin. In the previous section we talk about the virtues of the PCB purchased on Aliexpress due to its size, price and the possibility of being configured inside a box in order to even be able to build an electromyograph with the same dimensions as the MDURANCE one.

Regarding connectivity, it is currently very easy to find microcontrollers with wireless connectivity and there is also the possibility of configuring an Arduino Nano to send data via Bluetooth at a sampling rate similar to that of your product.

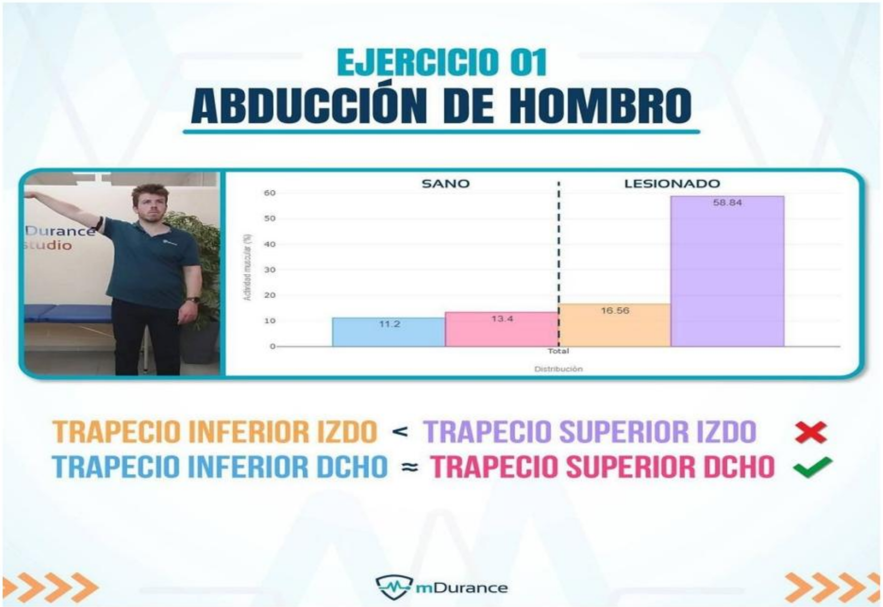


Figure 2.11: Trapezius activation in healthy versus injured patient [10].

Another interesting niche that can be seen in the market is the use of EMG as a human-machine interface, where it would be used as a method to control external elements or any type of machine by characterizing the user's muscle movements.

This principle is the same used for prosthesis control. A simple example of this would be found within what we know as Virtual Reality, where the aim is to immerse the person in a completely digital world and there would be an avatar controlled by the person's muscular movements. Video games are also an interesting method to introduce people to this type of technology and make interaction more friendly during experiments or rehabilitation. It is taught in a graphic and entertaining way what this is all about. Currently, on the Internet, you can find bibliography that deals with the topic and some developments, demos, etc. But none of these projects is notable or famous, it would be a scenario that is currently little exploited, but which could work very well and is used in this work [17].

Regarding the hand prosthesis market, more informal dissemination is needed to bring this technology closer to any type of public. Almost all the information refers to academic or scientific articles, which are not open access in most cases. On the contrary, on the “Biomakers” YouTube channel that we had mentioned in the previous section, they disseminate and teach this technology for free. You can even find material and courses on different branches of bioengineering. The only drawback of this could be that they do not usually go into depth about the developments and leave any information about code or designs to their courses (which are not exactly cheap).

It should be noted that they also have a web store where they sell material for the construction and start-up of robotic arms, sensors, material for 3D printing, etc. [18].

Alt Bionics [19], an American company created with the aim of building cheap prostheses, currently have quite advanced prototypes and are looking to open the market with a product that would cost around $3,500. To say that they are very active on social networks and support the community of researchers, aiming in the future to offer accessible dissemination on the subject.

It is impossible not to see the rise of robotics in recent years and this greatly benefits the construction of this type of implementation. Many companies in this niche work on the manufacturing of exoskeletons, robotic arms and offer the finished product or kits with 3D designs and assembly tutorials. This would avoid a lot of inconvenience, since it is an important part of the manufacturing for this type of system in which it would also be necessary to work on its software. An example of a company following this business model would be YouBionic [20]. Another option currently would be to download Open Source designs online or at a very low price for implementations of any type. With the possibility of adjusting them individually to a specific patient, this is one of the best options to build a first prototype.

Regarding the situation of these products in Spain, there is currently a national association [21] for people with this type of disabilities. The government has a program to offer free prostheses to amputee patients. Specifically, they are using the Michelangelo prosthesis [22], which returns a significant amount of movements. However, it is an implementation that, due to its price (around €60,000), is difficult for many people to acquire.

Finally, we would like to highlight that the therapists have indicated to us that not all people can access this free prosthesis program due to the existence of possible incompatibilities due to the conditions of the amputated area, adaptation to the limb and possible problems due to the sensation of phantom limbs. In our opinion, the existence of low-cost prostheses would allow more people to test whether they feel comfortable with them.

2.4. Conclusions

In this chapter, a review of different products, both hardware and software related to the development of the system that we will present below.

At the beginning of this project, we only had the Aliexpress sensor and it was later when we had access to the second Olimex board. Therefore, 2 configurations were made, always seeking to adapt to the needs of the moment. On the other hand, the summary presented in this chapter of the developments carried out by other authors has allowed us to establish certain characteristics of the applications that allow a development based on EMG to be used in areas other than the construction of prostheses. Thus, apart from the 3D prosthesis model, in this TFG the hardware part has been integrated with some well-known games.

Chapter 3

Developed Architecture

In the previous chapter we have made a general and informative approach about the current state of electromyography. We start, therefore, from a base with basic knowledge about biological principles, material for signal acquisition and information about developments or objectives that we would like to achieve in the future.

We will begin this chapter by talking about the technologies used in the development of the entire system. All this includes the programming part of Arduino and the development of the graphical interface to integrate all the modules of the architecture.

Figure 3.1 shows the schematic of the architecture whose modules will be described in the sections that we will present below: signal acquisition, signal processing and application. As an example, the results of some tests that were done to verify the correct approach of the system will be shown.

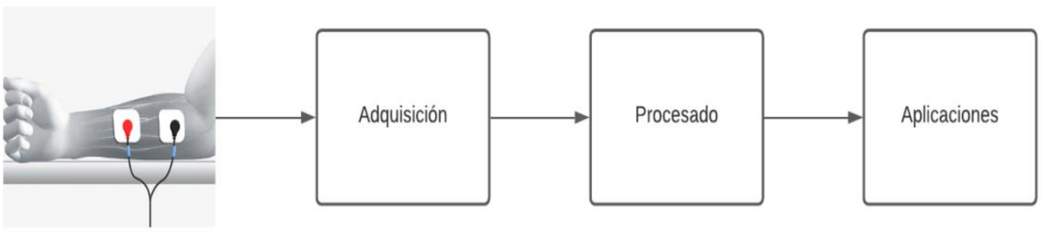


Figure 3.1: Architecture scheme.

3.1. Technologies

During this section we will comment and describe the technologies used throughout this work.

3.1.1. Arduino

Arduino [23] is an open source electronic platform based on easy-to-use hardware and software. It uses the Arduino programming language (based on Wiring) and the Arduino Software (IDE), based on Processing.

Over the years, Arduino has been the brain behind thousands of projects, from everyday objects to complex scientific instruments. A global community of creators (students, hobbyists, artists, programmers and professionals) has gathered around this open source platform. Their contributions have added up to an incredible amount of accessible knowledge that can be of great help to both beginners and experts.

It is essential to highlight the great importance that the existence of the community of developers that make up Arduino has for this project. Without the possibility of access to all the available documentation, for the use and understanding of the analog readers available on the board and serial communications, it would have been impossible to easily start development.

Arduino was born at the Ivrea Interaction Design Institute as an easy tool for rapid prototyping, aimed at students without experience in electronics and programming. As soon as it reached the broader community, the Arduino board began to change to adapt to new needs and challenges, differentiating its offering from simple 8-bit boards to products for Internet of Things applications, wearables, 3D printing and embedded environments. . All Arduino boards are completely open source, allowing users to build them independently and eventually adapt them to their particular needs. The software is also open source and grows thanks to contributions from users around the world.

3.1.2. Python

Python was chosen to develop the project. It was created in the early 1990s by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands, as a successor to a language called ABC. Guido remains the primary author of Python, although he includes many contributions from others. All versions of Python are open source. Historically, most versions have also been compatible with the General Public License (GPL) [24].

It is a powerful interpreted programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing make it an ideal language for scripting and rapid application development in many areas in most applications.

the platforms [24].

During the work, Open Source code libraries were used for the development of the application. The most notable were the following:

- Matplotlib.
- Numpy.
- Serial.
- Scikit-Learn.

Matplotlib [25] is a library that allows creating static, animated and interactive visualizations. It was created by John Hunter (1968-2012), who, along with many contributors, have dedicated time and effort to producing such a piece of software used by thousands of scientists around the world. In our project, it was mainly used to graphically represent EMG signals and their characteristics.

NumPy [26] is an open source project that aims to enable numerical computing with Python. This is why it was useful for us to create and manage the data structures that make up the muscle signals and thus be able to extract parameters (such as the average using) auxiliary functions that the library contains.

It was created in 2005, based on the initial work of the Numeric and Numarray libraries.
NumPy will always be 100% open source software, free to use for everyone and released under the liberal terms of the modified BSD license. It is developed publicly on GitHub, through the consensus of developers and the scientific community in general.
It also consists of a governance approach made up of several members in the form of a council.

Pyserial [27] is a library that encapsulates access to serial connections between two systems. Provides support for any version of Python running on Windows, OSX, Linux or IronPython. In our case, we use its functions to establish communications with the Arduino, synchronize it with our Python program and, finally, read the samples collected by the sensors. This process was very easy to implement thanks to all the available documentation on this project. It is completely free software and developed publicly on GitHub.

Scikit-learn [28] is an OpenSource library used for Machine Learning where we can find algorithms to train and evaluate models or preprocess data sets for later use. This project by David Cournapeau was born in 2007 during the “Google Summer of Code”. Years later, Matthieu Brucher would continue with it as part of his thesis project. In 2010 Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and

Vincent Michel of INRIA took the lead on the project and published the first publications in February 2010.

During this work we managed, thanks to its application programming interface (API) for support vector machines, to classify hand movements and use them as a human-machine interface. The library was in charge of training and validating the model, we simply adjusted some of the necessary parameters to obtain the best results.

3.1.3. Blender

Furthermore, for 3D modeling we decided to use the Blender tool [29]. It is a free and open source 3D creation suite. It supports the entire 3D construction process: modeling, assembly, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation. Advanced users use the Blender API for Python scripts to customize the application and write specialized tools (which we currently use in this TFG); These are often included in future versions of Blender. It is ideal for individuals and small studios that benefit from its unified and responsive development process. This tool also has a showcase where examples of many Blender-based projects are available.

It is cross-platform and runs equally well on Linux, Windows and Macintosh computers. Its interface uses OpenGL to provide a consistent experience. To confirm specific compatibility with your system, you can check the list of equipment tested by the developers.

As a community-driven project under the GPL license, the public is empowered to make small and large changes to the code base, leading to new features, sensitive bug fixes, and improved usability. Blender is priceless, but you can invest in it, participate and help in its development and project.

3.2. Signal acquisition

The first stage of the system shown in Figure 3.1 is the one corresponding to the acquisition of the signals. To read EMG signals, we have 2 configurations (see Figure 3.2): the first in which we use the Olimex boards and the second with the Aliexpress boards. In both cases, Olimex passive electrodes were used. As can be seen, the assembly corresponding to the Olimex plate (photo on the left) is much more compact and attractive. However, we did not have access to these sensors until late in development and therefore the first experiments were performed by the student with the configuration brought from Aliexpress.

As we mentioned in the previous chapter, Aliexpress sensors produce a positive analog signal that will be read by the analog-digital conversion inputs of the Arduino Uno. This board has an analog-digital converter (ADC) of the converter type due to its approximate transfers (SAR) [23]. This converter works by a simple binary search algorithm that uses a comparator to successively narrow a range that approximates the input voltage. In each step, the input voltage is compared to an internal reference produced by the Arduino itself. This value can represent the midpoint of the selected voltage range at each step of the process, the approximation finally being saved in a register. This ADC has a 10-bit resolution and the results will be ranged between 0 and 1023.

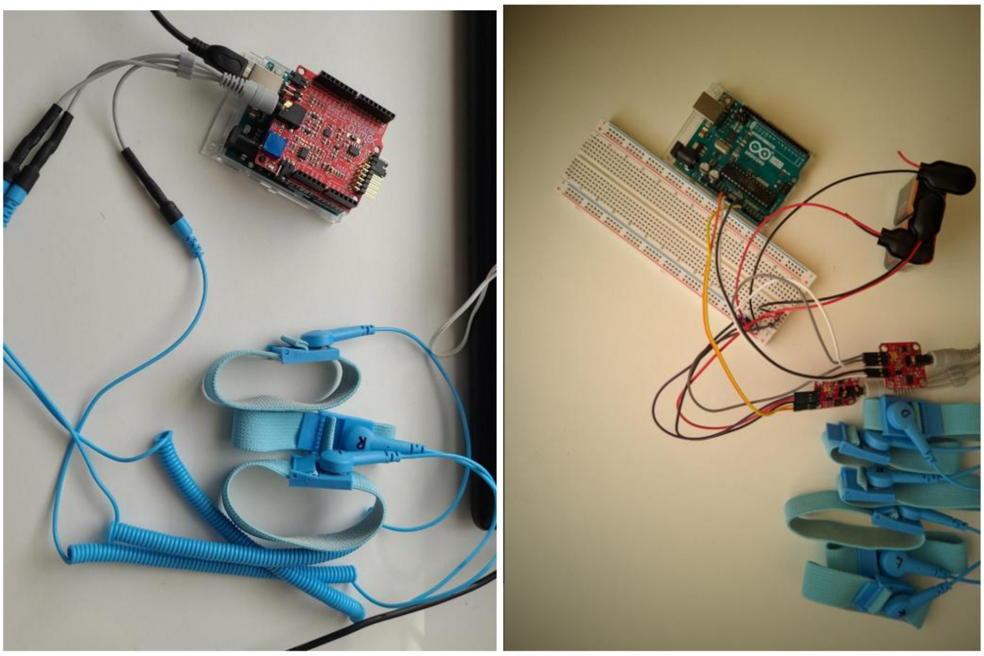


Figure 3.2: Board assemblies: Olimex (left), Aliexpress board (right).

In the previous chapter we have already talked about the assemblies of Figure 3.2, here we would like emphasize the parameters that are configured for measurements. For all experiments we set a sampling rate of 1000 Hz and BaudRate of 115200 baud to avoid a bottleneck in the transmission rate for the data. The reason for the first value is based on the following two ideas:

- The significant frequencies in the EMG are between 0 Hz and 500 Hz. Therefore, in order to work with this frequency range we must establish, at least, a sampling frequency of 1000 Hz, thus avoiding the aliasing effect [12].

- If you want to obtain good results in the prosthesis or in any other development, it is recommended that the response time of the application be less than 300 ms. Therefore, in our system we have decided to analyze sequences of 250 samples, which would correspond to an approximate response time of 250 ms [4].

It is important to keep in mind that, although we look for good results with 250 samples, it is advisable during the experiments to capture many more to have plenty for the analyses.

For the final measurements, a software tool was created in Python that is responsible for connecting to the Arduino, synchronizing with it, reading 1000 samples of up to 2 channels and exporting everything to Comma Separated Values (CSV), a standard format that will be useful to feed the analysis libraries or classification models that were created. In addition, it offers a simple graphical interface in which you can manage the type of measurement we are doing, keep track of them, show the result graphically and even interact with the user using a sound to indicate when to perform the movement.

3.2.1. Preliminary tests

To illustrate the operation of this stage of the architecture, we will show the results of several tests carried out by the student with the Aliex press sensor configuration. We then remember the 4 states of the hand that we intend to identify:

- Wrist flexion.
- Wrist extension.
- Repose.
- Cuff closure.

Figure 3.3 shows the signals obtained in the two channels for three movements: wrist extension, wrist flexion, rest and fist closure. This is a figure in real time. The blue signal corresponds to the first channel located in the flexor carpi radialis muscle and the red signal to the second channel, located in the extensor carpi radialis longus muscle.
It is important to note that the scale adapts to the signal, so the rest subfigure has a much smaller scale. We remember that the Arduino ADC has a resolution of 10 bits and therefore at most our signals will reach 1023 units.

We easily see how for wrist extension (a) the amplitude of the red wave is much greater than the blue wave, and vice versa, during flexion (b). Placing the electrodes on muscles that are antagonists allows us to make a spatial classification of the movements. For rest (c), we have peak values in the 2 signals that do not exceed 240 units, Even in the flexor we see values very close to 0. On the contrary, during fist closure (d), the 2 sensors reach maximum values close to 1023.

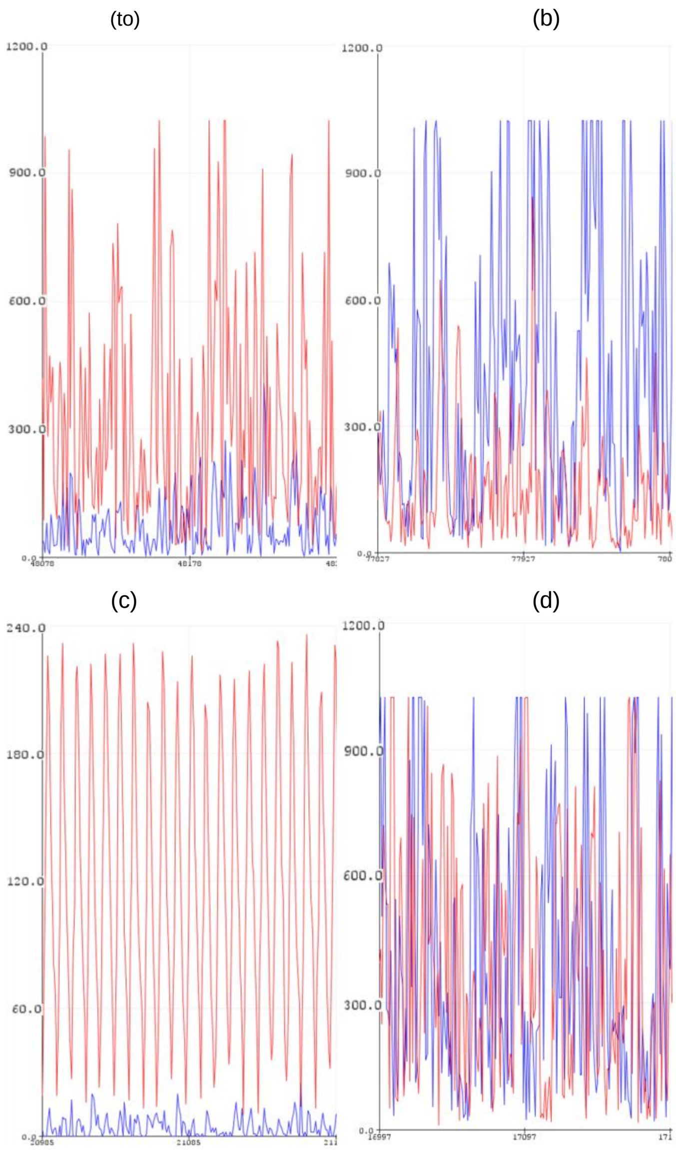


Figure 3.3: Serial Plotter output in the Arduino IDE for (a) extension, (b) flexion, (c) rest and (d) fist closure.

Looking at this first window of samples it is easy to verify that our approach is correct and that the 4 states can be easily distinguished.

3.3. Signal processing

Once the signals have been acquired, the next module in Figure 3.1 is the processing module. This module is responsible for extracting certain characteristics from the signal obtained that allow us to determine the movement made with the muscles. Therefore, this processing will be divided into two main tasks: feature extraction and classification of the movement performed. To do this, the techniques that we explain below have been implemented.

3.3.1. Feature extraction

The EMG signal obtained and that we saw in the previous test is an example of what is known as Raw EMG. The raw signals contain essential information, but in a format that is not useful for detecting the type of movement performed, since, as we mentioned during the introduction, it varies between people and even varies depending on the position of the electrodes. It is very common that the next step then consists of rectifying the signal, that is, obtaining the absolute value of the wave.

Rectification can be of two types: half-wave, in which the negative values are discarded and only the positive ones are taken into account, or full-wave, in which the absolute value of the negative values is obtained. In our case, full wave rectification is preferable for the subsequent analysis that we apply. In the scientific literature, EMG rectification has also been used to more easily detect low frequencies (indicative of the firing frequency of the motor units), since these are not easily detectable [30]. Although this statement is true, there are cases in which this technique can cause peaks to be detected at low frequencies that are not related to the discharge rate. Ultimately, this will be useful to us, for example, to analyze the amplitude in the time domain.

In the EMG signals captured by the electrodes we find random components caused by background noise in muscle activity, caused by the organism itself, the placement of the electrodes or even by the measurement equipment itself. Therefore, its amplitude cannot be defined with specific characteristics of the waveform (such as the peak value), but needs statistical estimations. To calculate the EMG amplitude, two methods can be used: root mean square (RMS) or rectified mean value (ARV). The RMS is the most used method due to its clear physical meaning, while the ARV simply provides the measurement of the area under the EMG signal [31].

The RMS is a statistical measure that seeks to collect information in a series of discrete values or mathematical function of a discrete variable without affecting the sign of its components, that is, in a certain way we are simultaneously applying the concept of wave rectification when calculating this feature. From N samples of a signal

RMS is calculated by doing the following operation:

$$\mathrm{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} x(n)^2}, \tag{3.1}$$

where x(n) is the nth sample of signal X.

Therefore, we consider that RMS is the most suitable method to extract the features of our system. In this way, this RMS value will be used to determine the movements performed by the user. That is, from each sequence of 250 samples its RMS value will be extracted to classify it into a type of muscle movement.

3.3.2. Threshold based classification

Having now understood the concept of RMS, this section will explain what its function will be and where we will apply it. The idea is to recognize the moment in which muscle contraction occurs, which represents the basic principle for the existence of movement, and differentiate it from rest. Using only 1 sensor, these 2 states of contraction or rest depend directly on the area in which we place the electrodes. In our case, having them on the forearm will cause a contraction when performing any hand movement, such as closing the fist, flexion or extension of the wrist, or even during certain movements in the fingers.

Although the threshold-based method can be used to differentiate two states which want, in this project we have applied it to classify the following:

- Wrist flexion.
- Repose.

To recognize these 2 movements we will use RMS, since during muscle contraction the motor units are fired and the amplitude of the electromyographic signal increases considerably. Then we can set a threshold value with which we can differentiate whether a movement is being made or whether it is at rest. This threshold value is determined during a first training stage, where the user performs test measurements to calibrate the system.

The calculation of the threshold is very simple. In our case, we will use several independent measurements of the EMG signal during rest and during contraction. From the samples obtained in each measurement, we will calculate the RMS and then calculate the arithmetic mean for each state. Finally, the midpoint between these 2 values would work as a threshold.

Based on our experiments, we can say that 10 measurements of each state are enough to obtain good results. Increasing their number failed to offer improvement and, furthermore, hindered the experiments by lengthening the sessions.

With this classifier it would be possible to control a hook-type prosthesis, where contraction (wrist flexion in our case) or increase in RMS would mean closure of the hand during implantation. In this case, the calculation of the threshold automatically has been explained, but it is important that the system also allows manual calibration of it, since we may encounter patients who have difficulties reaching said value, either due to muscle fatigue, ailments in the extremity, etc.

3.3.3. Classification using support vector machine

The threshold technique is easy to program and works well if only 1 sensor is used. It is clear that the operation is very trivial, it does not have any type of complexity, but what would happen if more sensors were used? How do we manage multiple thresholds?
Well, in this project we get to work with 2 sensors as we mentioned in previous chapters. Therefore, now instead of just flexion and rest we have 4 possible states, which from now on we will call “classes” of movement:

- Wrist flexion.
- Wrist extension.
- Cuff closure.
- Repose.

In a simplified way, we can say that the idea lies in a spatial classification where due to the placement of the electrodes and the nature of these movements, during the flexion of the wrist one of the sensors would be “activated” while the other would remain in place. state of rest and, likewise, during extension the opposite would occur. At rest, the 2 sensors would be inactive and, during fist closure, both would be activated.

To model this behavior we used several measurements of each class (in our tests, there were 15) and the classification algorithm based on support vector machines (SVM) and the RMS characteristic calculated on each sensor. This algorithm draws a vector space with all the measurements of each class and draws planes (or hyperplanes) to delimit areas between the movements, using the “Support Vectors” (reference those points that are closest to the border with the other class).) to draw these areas in which features of the same type are grouped. To do this, it uses different kernels that will shape the areas according to the nature of our data. Here we will study the Linear, Polynomial, Radial and Sigmoid kernel[32].

It is also supported by a training process, where it checks based on a set of test points whether the drawn areas are correct when classifying this test data and thus, calculating how effective the model is.

The code developed in this TFG is quite simple. The Scikit-Learn library was used to train and validate the models. Add that this type of classifiers perform a binary classification by default, so it is necessary to add small tricks to have several classes of use. It builds n_classes ÿ (n_classes ÿ 1)/2 classifiers, trains them and applies a decision function of the type 1 against 1 (algorithm that can later be transformed to 1 against the rest where the number of classifiers decreases until it reaches the number of classes).

3.3.4. Preliminary tests

During this section, the tests carried out with both the threshold technique and the SVM machine will be attached. We remember that these first tests were carried out by the student with the Aliexpress configuration.

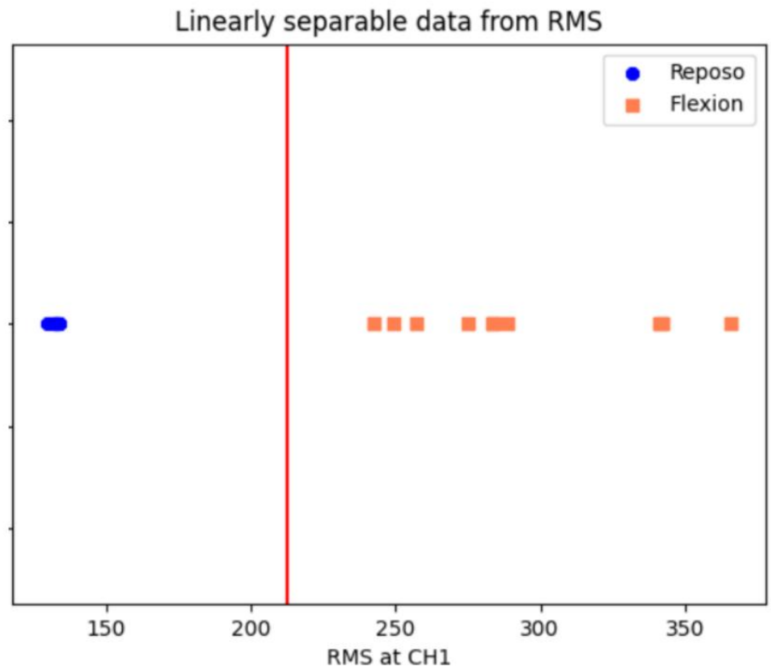


Figure 3.4: RMS calculated on the measurements made for the threshold technique: rest (blue), flexion (orange) and red line the calculated threshold.

In Figure 3.4 we wanted to represent the training results for the threshold technique described in section 3.3.2. The RMS value of each measurement can be seen on the X axis of the figure. The blue cloud corresponds to the 10 measurements at rest and the 10 orange ones are those obtained during flexion. Finally, the red line located at a value of 210 is

This is the threshold calculated by the system that will allow us to make the classification between the 2 movements. This value was later used during the game.

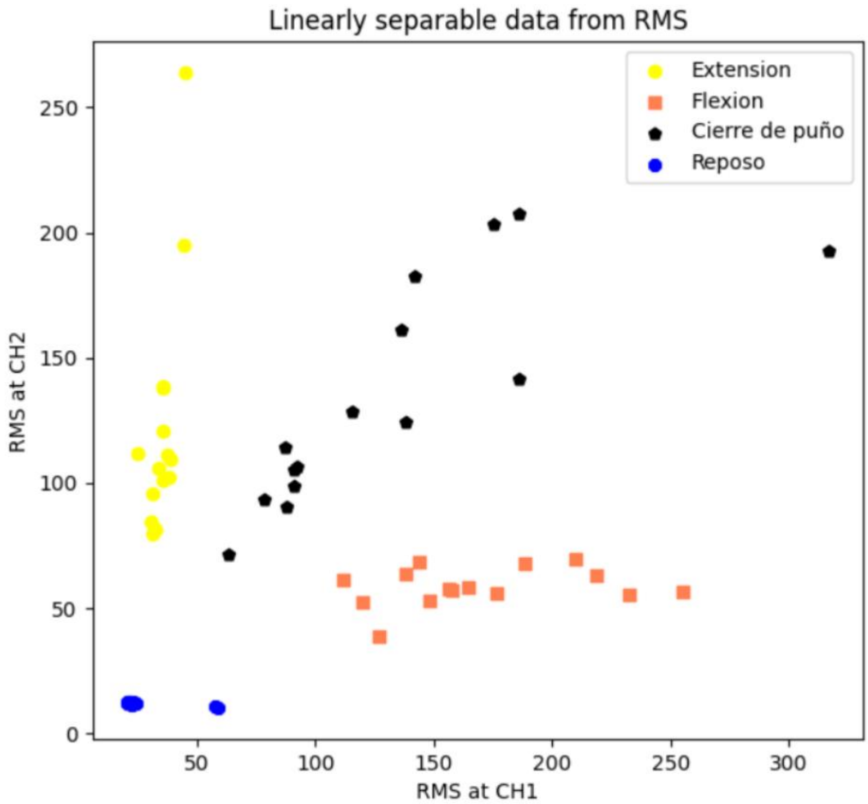


Figure 3.5: RMS for each of the four movements: rest (blue), extension (yellow), fist closure (black) and flexion (orange).

In Figure 3.5 we have a 2-dimensional representation of the RMS for the 2 reading channels and 15 measurements in each class. The blue group of points corresponds to rest, black to fist closure, orange to extension and blue to flexion. As can be seen, the data for the measurements show what we have been saying: the spatial classification is effective and the RMS values during contractions are sufficient to differentiate movements or groups of points well. For this case, a simple classifier might be able to easily identify each of our classes.

Looking at this map we can now understand the output of our classifier. Figure 3.6 corresponds to the points in the previous figure and its trained model. Each colored zone represents a class in which any point that falls within will be labeled as the same. These results are also a good option to see graphically that the Linear and Polynomial models are the ones that have best managed to define and group the movements, while the other 2 nuclei do not fit the nature of our data at all.

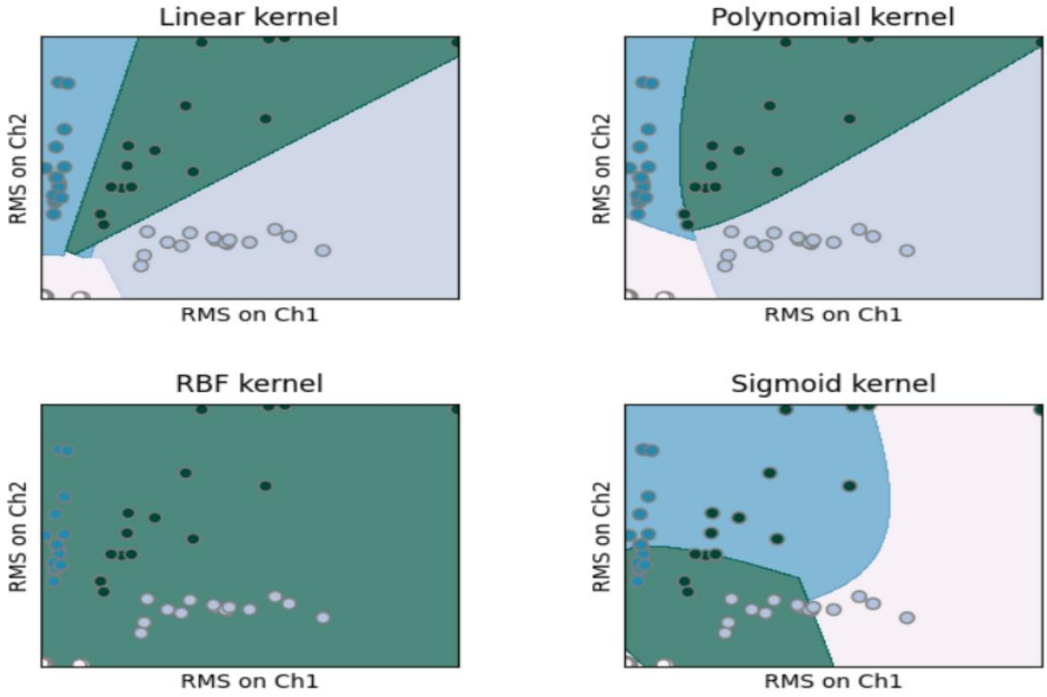


Figure 3.6: Classification map for different models.

3.4. Applications

At this point, the developed system can already read muscle signals, process them and recognize the movements that the user makes. In this way, in our work we have integrated this system with two applications that can be controlled through muscle activity: a simple video game and a 3D model of a forearm prosthesis that is capable of following the recognized movements in real time.

To facilitate interaction with users and improve the usability of the system, a general-purpose graphical interface was developed that allowed access to its different functionalities, including the execution of these applications. This interface also allows us to work with the EMG signals in an intuitive way, allowing us to analyze the extracted data or the raw signals.

Figure 3.7 shows the main interface, as you can see, starting from the top to the left, we can:

- Insert the name that we want to give to the measurement file. He will also be in charge of keeping track of them. That is, if several measurements are made for the same user, the system is responsible for listing them.

- The “MAKE MEASURE” button allows measurements of up to 2 channels. It has a simple user interaction through a sound system to indicate the start of data capture (we will see more detail in the next chapter). Furthermore, once the measurement has been carried out, it shows us the results and allows us to discard it in the event of an error.

- The next “REAL TIME PLOT” button draws the EMG signals in real time.

- By clicking on “ANALYZE DATA”, the system analyzes all the measurements that were made during the session for a user. To do this, save them in a labeled folder.
  In addition to the latter, it calculates the threshold between rest and flexion.

- You can also manually define a threshold for the connection with the game or 3D model.

- Run or stop the game or the 3D model by pressing the buttons: “PLAY!”, “CONNECT 3D!”, “STOP!”.

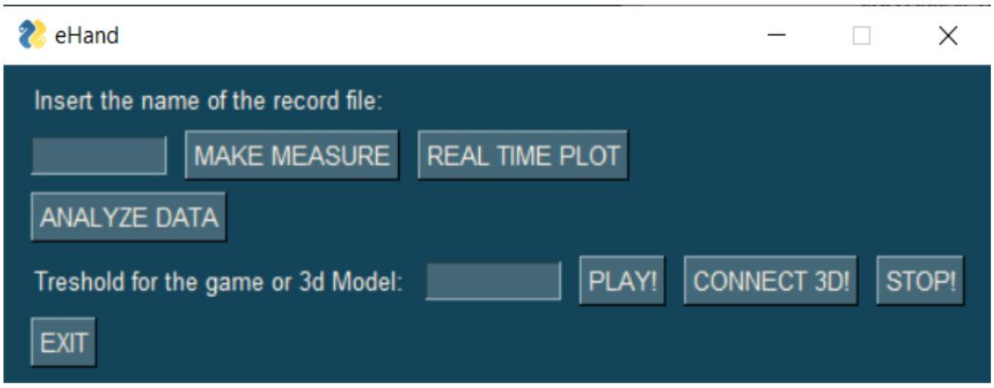


Figure 3.7: Application interface.

Below we will detail the two applications that can be controlled through our EMG system.

3.4.1. Jumping video game

With the threshold method explained before and our application, the user can currently interact with a game. Figure 3.8 shows an example of what is achieved with this interaction. It can be found on a well-known website [33] and for the experiments we will use one of the levels in which the user must make a character jump at the right moment to not lose.

The system manages to interact with the game by detecting flexion in the wrist. Simply, when this happens, it executes the action of a "click" of the mouse as if it were another peripheral connected to the computer.

The idea allows us, in addition to making the character jump, control any computer application. This is the versatility we are looking for so that the threshold technique or SVM machine can be used in any other game or human-machine interface.

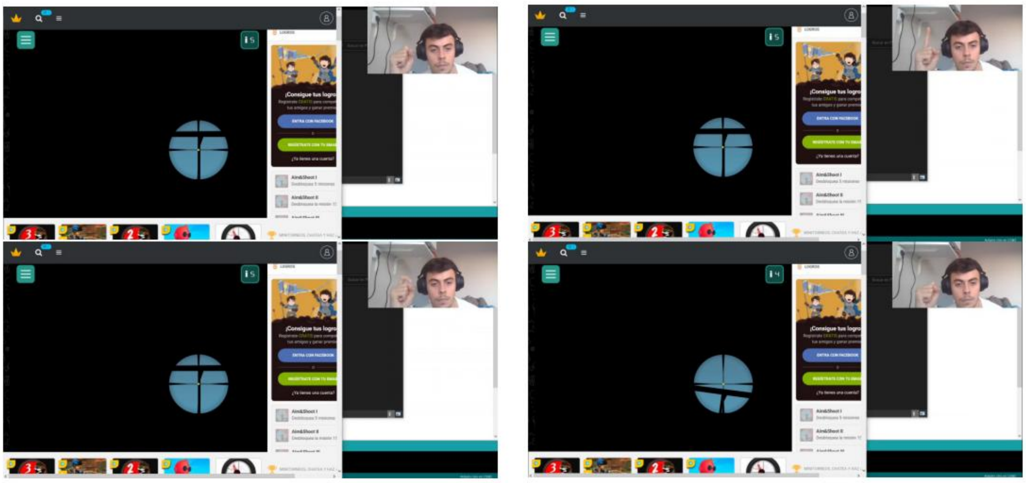


Figure 3.8: Example of EMG as an interface for games.

3.4.2. 3D prosthesis model

The second application developed was the 3D model of a forearm prosthesis controlled through our system. This was built in Blender, a tool that also allows the execution of programs in Python, which allowed us to connect it to the user's EMG and follow the person's movements in real time.

The code is abstract enough to be used using the threshold method or the SVM machine. In the first case, the model will simply open and close the hand, while in the second it will also flex and extend the wrist. Figure 3.9 shows the 3D model of said prosthesis opening and closing the hand.

Currently, we have 4 movements mapped that we could think correspond to what the prosthesis should make. This approach is completely correct, but then we are left with only 4 actions out of the many that a real hand has. Therefore, it would not be bad to move away a little from the concept of 100% mimicking each of the states that exist and use the few resources we have to achieve more.

Currently, this is already done in prostheses through the concept of commands where some of the trained movements are used to function as a wild card.

By combining these and the current state of the machine, more than 4 actions can be coded in our system. For example, if we use the fist closure as a wild card, the user can change between different grip positions. Finally, through extension and flexion you can regulate the closing or opening angle that you want to give to that position. Another example could be using the fist clench to alternate between different movements. In a first state, closing and opening of the hand would be controlled. During the second state, the rotation of the forearm would be controlled. The amount of opening, closing or rotation would be controlled by flexion and extension of the wrist. As a note, say that this type of wild cards can also exist in physical form, a simple button on the prosthesis that switches between different possibilities.

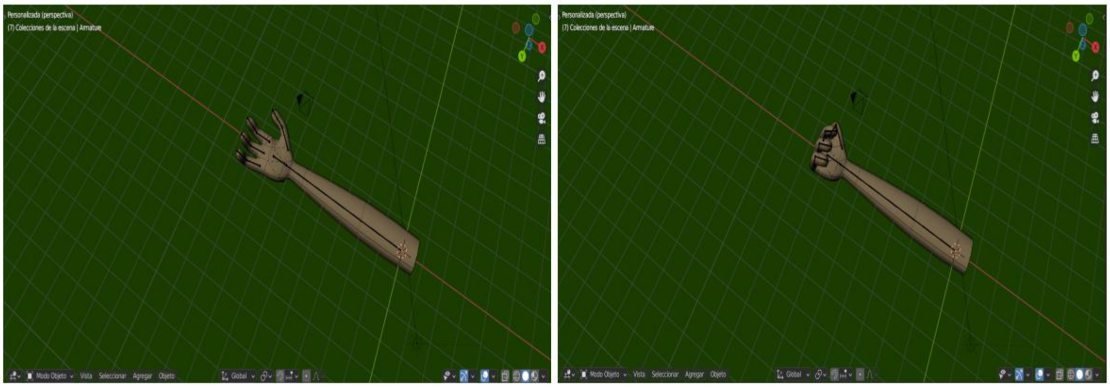


Figure 3.9: Example of fist closure in the developed 3D model.

3.5. Free access code

As indicated above, the developed code is fully available in the Github repository, eHand [16]. This is a repository that is constantly changing and subject to improvements, but for now, the directory structure is mainly divided into 4 parts:

- The first will be the “doc” directory where all the documentation related to the project will be found.
- We continue with “hardware”. Here is all the code that controls the microcontrollers and sensors, it is also responsible for communications with the application.
- “Measures” and “dataset” folder. They contain measurements and data sets used for analyses.
- In the following directory, “python”, there is all the code related to the development of the applications. In this structure we can find:

- The “eHand” folder contains the main application, graphical interface and respective calls to the different modules.
- The rest of the directories then correspond to the different modules or libraries that will be used by the main application.
- Within these directories, we have code for measurements, analysis, models classification and even demos for the game or 3D model.

First of all, the Arduino must be configured using the code available in the “hardware” directory: “olimex_emg_reader.ino” corresponds to the configuration in Figure 3.2 on the left and “analog_emg_to_serial.ino” to that on the right. When you run it, you only need to send the word “ini” through the Arduino Serial Plotter tool to display a real-time graph of the signals that the devices are capturing.

Subsequently, we will execute the file “eHand.py” from the “python/eHand” directory. This will open the interface shown in Figure 3.7 which, as indicated above, allows you to execute all the functionalities: take measurements, analyze the data, play and control the 3D model.

Likewise, we will update the “README” file in the GITHUB repository with all the information necessary for the use and manipulation of the code.

3.6. Conclusions

In this chapter the modules of the architecture developed in the TFG have been described. The first module consists of signal acquisition using electrodes and an A/D conversion system. The second is the extraction of features and their subsequent classification using thresholds or support vector machines. Finally, the interface that allows signal acquisition, training and running two applications has been shown: game and 3D model of a prosthesis.

Chapter 4

User testing

At this point we have already managed to develop a system capable of reading, processing and applying EMG signals for the control of human-machine interfaces. Now is when one of the most important questions arises: Will it work and will good results be obtained in users who do not know anything about EMG? To answer this question, different tests of the two configurations were carried out with users outside the development.

We dedicate this chapter to describing the experience of testing applications with users without any knowledge or prior use of this type of technology. We will talk about certain practices that were crucial to the success of the experiments.

4.1. Test subjects

We will have 5 people for the tests (see Table 4.1): a middle-aged woman, 2 boys and 2 young girls. For the last 2 users we were only able to perform the threshold tests. In the rest of them we were able to do experiments for all the techniques.

None of them are the student or the directors of this TFG, therefore they are unaware of the development and the results that are expected to be achieved. The configuration with the Oli mex plates will be the one we use for these tests. The assembly is much cleaner, allowing a good impression on users who might feel uncomfortable seeing the wiring and installation needed for the other PCBs. Furthermore, it will be interesting to see the comparison that arises between this and the previous experiment carried out during chapter 3. The placement of the electrodes and configuration of the software is the same as that described in the previous chapter.

In Table 4.1 we add information about the physical build for each subject. We remember that during chapter 1, it was explained that the capture of superficial EMG signals can be conditioned by the physical characteristics of the person in question. This could be your age or body build. We tried, therefore, to have tests with various age ranges and different existing mesotypes (see Figure 4.1): endomorph, ectomorph and mesomorph.

Name Age Build		
User 1		ectomorph
User 2	56 23	mesomorph
User 3 ectomorph 20		
User 4 22 endomorph		
User 5 21 mesomorph		

Table 4.1: Subject information for the tests.

Figure 4.1: Mesotypes of the human body.

4.2. Protocol

Before talking about results, we believe it is important to detail several aspects to take into account when working with EMG signals. In principle, it was expected that the achievements would be similar to those of the preliminary tests, since biology does not lie: during a muscle contraction there will be an increase in power in the electrical signal of our body and, therefore, we had the advantage of knowing what effect we hoped to obtain.

In the session, users receive an explanation about how the system works and how EMG signals are captured. All of them indicated their fear of receiving some type of electric shock from the electrodes or even feeling some current through them as if it were electrostimulation. Therefore, it is also explained to them that the laptop with which the measurements are recorded is disconnected from the power and that the experiments will not cause them any type of damage or discomfort. If they feel uncomfortable, they can abandon the tests at any time they wish.

Once you calm them down and explain what the procedure is going to be, it is very important.

arm placement during measurements. The flexors, extensors of the wrist and, in general, those areas of the forearm, contain muscles that can be active without looking for it and affect the measurements. Therefore, it is necessary to always start from a relaxed and neutral position, where any rotation of the forearm or wrist is left out of the equation.

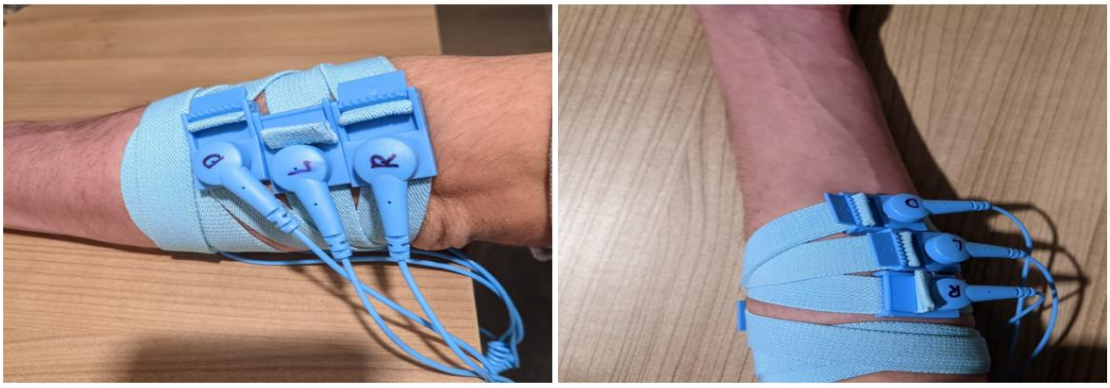


Figure 4.2: Placement of Olimex dry electrodes. Extensor on the left, flexor on the right.

To start, it is advisable to first place and capture signals from only 1 sensor.
In order for all users to use the same configuration and have the same conditions, in our experiments with the threshold technique, the electrodes were always placed on the forearm flexor, as in the preliminary tests. Figure 4.2 (right) shows such placement. With this, measurements could begin to test the threshold method and control of the video game. Even so, at the beginning of the sessions it is a very good practice to begin by reviewing the status of muscle activity using the Serial Plotter of the Arduino IDE or from our tool, which gives us a first look at the correct placement of the electrodes.

Next, they are asked to do the wrist flexing exercise or the fist opening and closing exercise consecutively to, if the placement and configuration is correct, check the increase and decrease of the amplitude in the wave between the resting states. and contraction. Regardless, during wrist flexion or fist closure, this muscle area always contracts.

If you do not see any fluctuation in the signal or notice that the returned values do not correspond to those previously seen, one of the following two situations may possibly be happening:

- Electrode placement is incorrect;
- The physical conditions of the muscular area are preventing the correct reading of the signals (high percentage of fat or low muscle mass).

If this is the first problem, we remember that it is always important to ensure the correct placement of the electrodes and also play with the separation distance between them. This is why we decided to use dry electrodes that allow these types of errors to be easily corrected. If wet electrodes are used, they should be removed, discarded the patches (if they are damaged) to take new ones and try again with another positioning.

If it is the second cause, here we will be very conditioned by the hardware we use. With Aliexpress boards, we will not have the possibility of adjusting any type of gain to increase the sensitivity of the equipment. On the other hand, with Olimex plates we have the possibility of physically adjusting the gain of the equipment. It must also be said that in our case, we have not suffered this problem in any of the tests with either one plate or the other.

Continuing with the session, at this point it is also normal to see, in the face of ignorance, how the users force the fist to close and use more force than necessary, potentially causing fatigue in the area. To solve this problem, it is explained to them that it is not necessary to exert excessive force and that, if they control the movement better, better results will also be obtained. It is more effective to make quick contraction movements, like when we drop something from our hand and, by reflex, we try to catch it in the air.

Once it has been verified that the electrodes have been placed correctly and that the system adequately captures the signals, we can now proceed to make the first measurements.
We remember then that with 10 recordings of rest and contraction we can already calculate a functional threshold for the game. Everything is done through the graphical interface of our eHand application and, so that you understand what the process is going to be like, we will start with the rest readings, where you do not have to interact in any way or make any type of movement, thus being able to familiarize yourself. with the course of sampling.

The application has been developed so that during this training phase, you hear a significant beep that lasts 2 seconds. It is explained to them that if they have to make any movement with their hand it would have to be at the precise moment when they stop listening to it. In these first measurements, since they are in the resting state, you do not need to do anything.

Once the rest tests have been completed, we move on to capture those corresponding to wrist traction/flexion. To do this, as mentioned, the application will use an audible alert, indicating to the user when they should perform said movement. The push-up should be performed right after the audible alert ends. The system will capture samples for 1 second, or what amounts to 1000 samples with the settings used.
We remember that it is always better to have plenty for analysis and this number of them seemed significant enough to us.

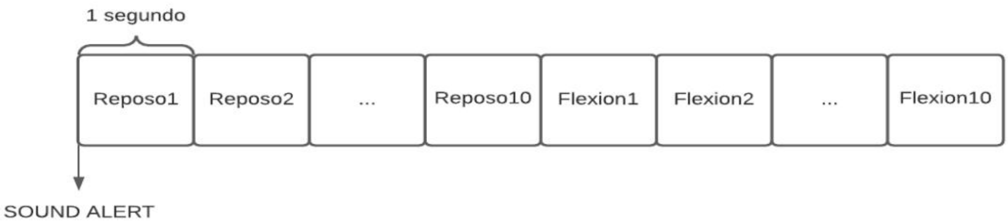


Figure 4.3: Scheme of the pattern followed in measurements for the threshold technique.

After this, it will display a graph with the captured signals on the screen. Through this action the student can check the signals and decide if a good measurement has been made or not. We would repeat this process described in Figure 4.3 until we have all the captures made. In general, it usually takes between 10 and 15 minutes, depending on the quality achieved during the recordings and if it is necessary to repeat any of them.

The technique of making a sound before each capture is a little attention trick for when contraction or movement measurements are involved. We make these quick and meaningful. Thus, we ensure that we obtain the best results. Finally, once this is done and the threshold calculated, we can move on to one of the applications.

4.3. jumping game

Once the training phase is finished, we proceed to test one of the applications: the game. Figure 4.4 shows a photo taken during the session.

As mentioned in the previous chapter, the objective of this application is for the user to be able to control a video game by flexing the wrist. It consists of making the main character jump to avoid a series of obstacles throughout a map. If you manage to jump all the obstacles, you pass the level.

In each experiment, time is dedicated until the person is able to overcome it without difficulty. In less than 10 attempts they achieve it, they usually spend between 5 and 10 minutes on it.

During the measurements, the user may not have fully understood the essence of the system and how to interact with it, but once the game is started, this perception changes. Direct interaction with something as simple as making a character jump by simply moving their hand makes them better understand how the system works and even helps them master this type of interface.

In order to compare the results obtained with the users, Figure 4.5 shows the RMS values calculated during the training phase for three users.

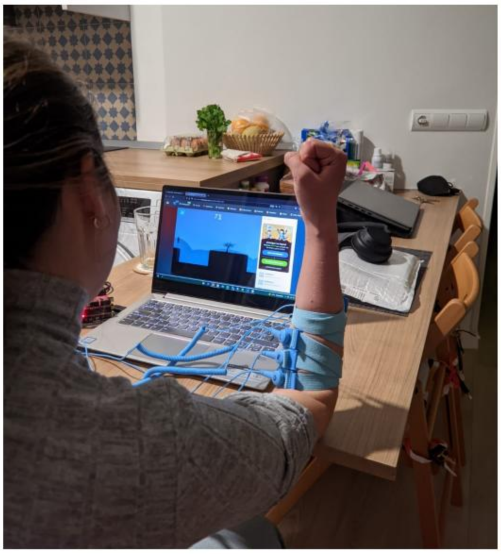


Figure 4.4: User testing the game.

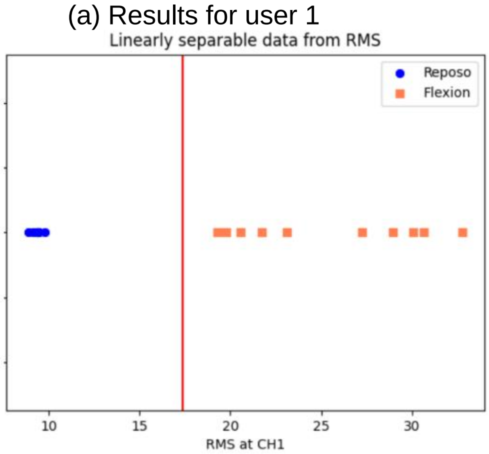
The obtained threshold is also shown (red line). The flex values (orange) are what significant enough to be able to make the classification. What's more, we can see than the measurements at rest (blue), remain stable in terms of RMS and form a very consistent point cloud, thus allowing the calculation of a threshold (red line) that separates correctly the 2 zones of movement. Table 4.2 shows the calculated value for the threshold of each of the users. As can be seen, there is a big difference between some thresholds, indicating the importance of calculating the value for each user instead of using a predetermined one.

The game consists of 7 jumps and the difficulty lies in the fact that users must perform the action at the right time to pass the level. In general, the results during This test was satisfactory in all cases: those who tried it later achieved After some attempts, you can pass the level without any problems. Table 4.2 shows the number of attempts they had to make to achieve the objective.

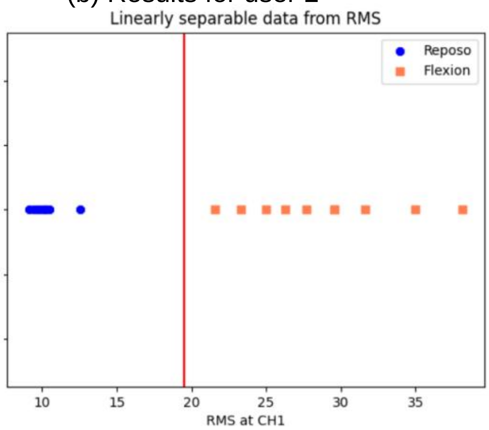
	Threshold	Attempts Game
User 1	17.3	5
User 2	19.4	3
User 3	25	4
User 4	4.5	6
User 5	4.4	4

Table 4.2: Threshold and number of attempts required to pass a level of the game for each user.

(a) Results for user 1



(b) Results for user 2



(c) Results for user 4

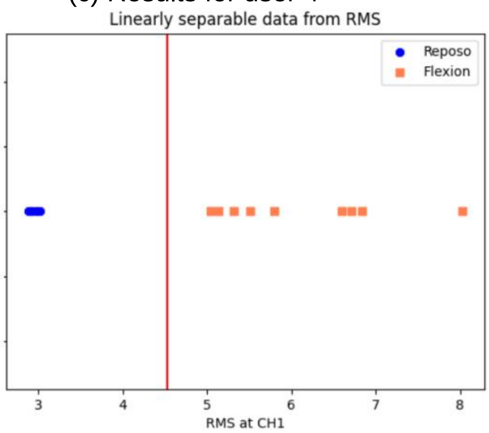


Figure 4.5: RMS results (in order) for the different users: rest (blue), flexion (in the range), red line the calculated threshold.

4.4. 3D model

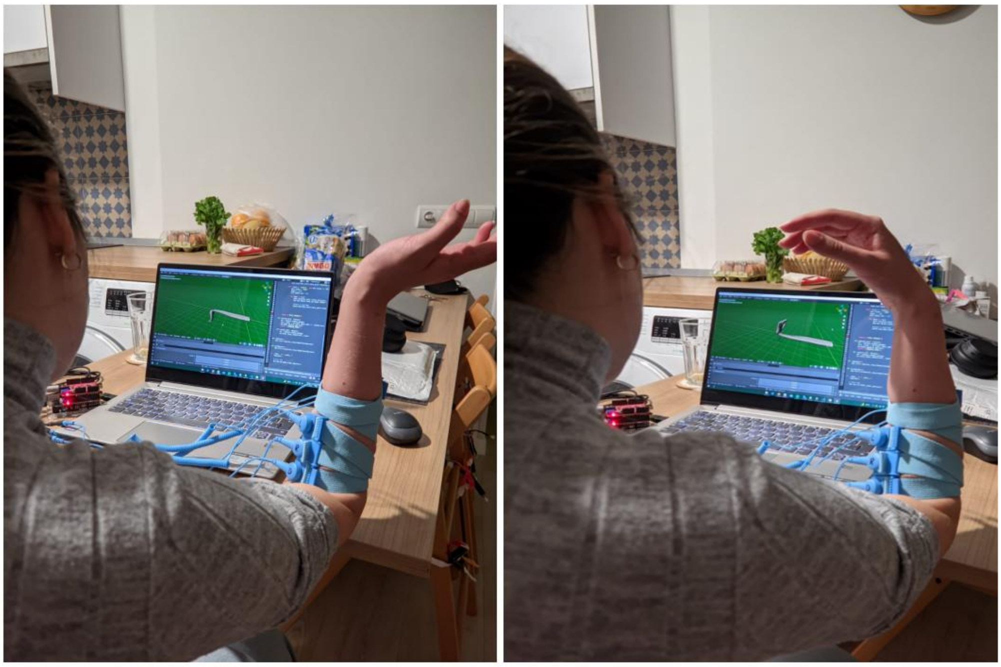


Figure 4.6: User testing the 3D model.

Having been successful with the game, you can test the 3D model (see Figure 4.6) with the threshold method or, go directly to placing the second sensor to begin with the measurements of the other classes of movement necessary to train and build the SVM model.

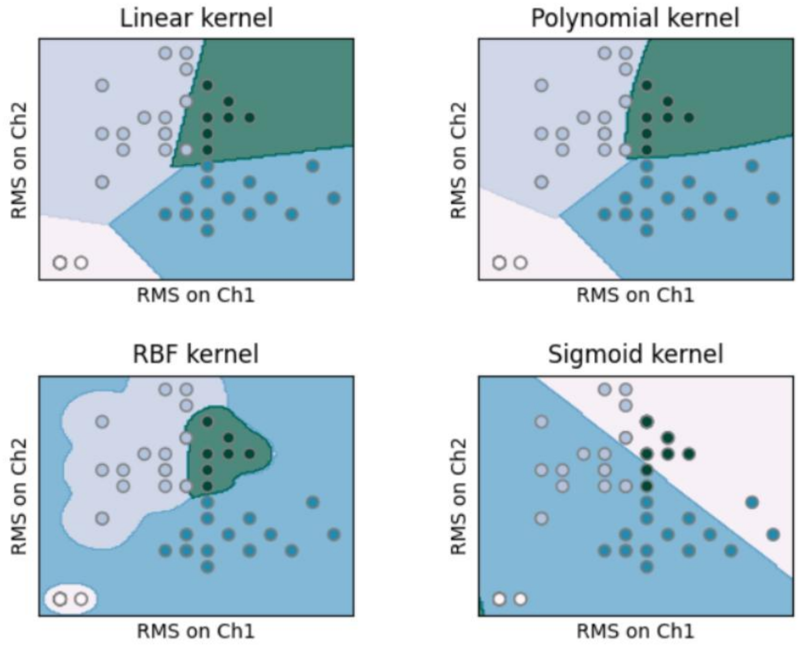
Since the person already knows the procedure and understands the situation better, this process would be done quickly. We remember that for the correct training of the machine, we will use 15 measurements of each class. The experiment would last approximately about 20 minutes.

As an example, Figure 4.8 shows the screen in the Blender application, where you can see all the types of movement that have been characterized in this work. We have, starting from the top left, rest, closed fist, wrist flexion and extension.

This section will be dedicated mainly to the results that were achieved during the training and validation of the SVM machine for the first 3 users. This part is important in the future to be able to implement a prosthesis with sufficient intelligence to recognize various movements.

As an example, Figure 4.7 shows the maps obtained for users 1 and 2 with four different cores. As can be seen, the linear and polynomial kernels perform a better separation of states for the two users.

(a) Results for user 1



(a) Results for user 2

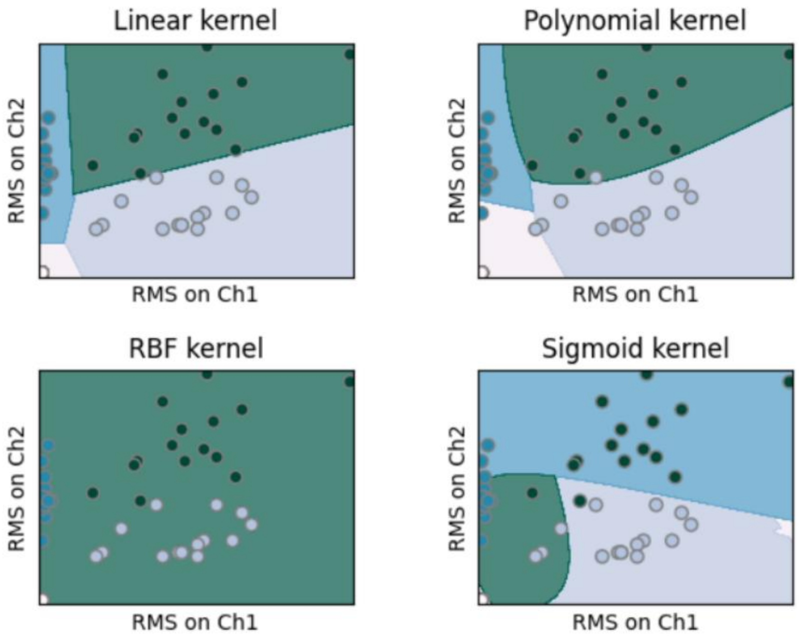


Figure 4.7: Classification maps for users 1 and 2.

To validate the correct operation of the SVM machine, k-fold was used as a statistical method to estimate the classification ability achieved with the data obtained in the measurements. K-fold [32] consists of cross-validating a model, dividing and “folding” the data sets at different levels that will be used to train and test k models in a mixed way. This is a resampling procedure used to evaluate machine learning models with a limited data sample.

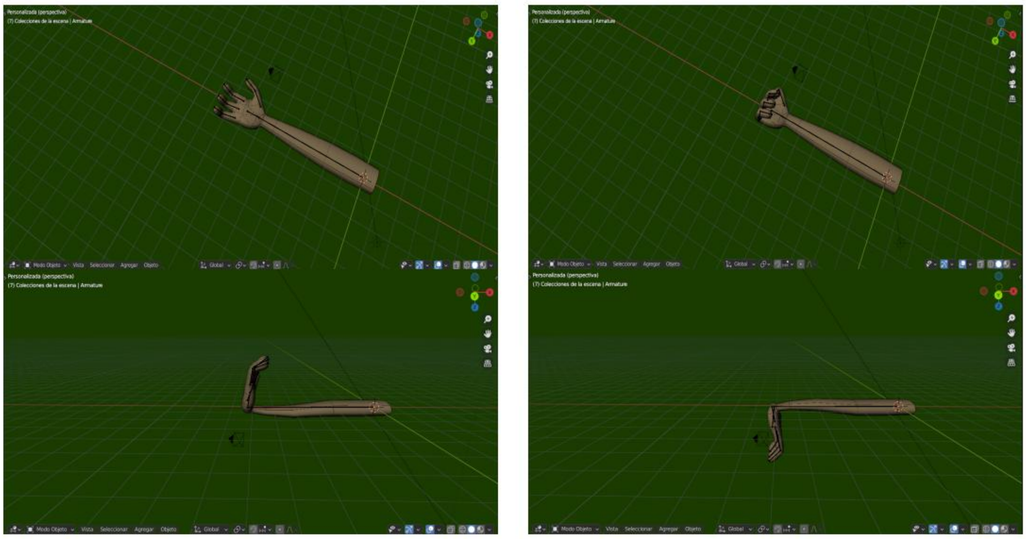


Figure 4.8: Example of all the movements achieved in the 3D model.

As such, the procedure is often called k-fold cross-validation. For example, when k = 10 this reference becomes a 10-fold cross-validation. In our case and due to the size of our data, during all experiments we will use 3-fold cross validation.

This method is mainly used to estimate the skill of a model with unknown data. That is, we use a limited sample of data to estimate how the model in general is expected to perform when used to make predictions about another set that is not used during training. The algorithm is popular for its simplicity and for providing unbiased results, since it is responsible for training and evaluating the model with all the available data. The general k-fold procedure is as follows:

- We shuffle the data set randomly.
- We divide the data set into k groups.
- We take one of the groups as the test data set.
- We take the remaining groups as a training data set.

- We train and evaluate.
- We repeat this process until we have evaluated the model with each of the unique groups that were formed at the beginning.
- We extract the skill of the model using the scores obtained in each iteration.

Finally, once the different models have been validated using k-fold, we can decide, Based on the score, which one will be used in the final development of our system. The model that obtains the best precision results will be trained with the entire set. of data available. This model will be the one used to test the application of the model 3D in end users.

The results for the k-fold validation for each user and core are found in the Table 4.3. In all results, the success value is represented by a real number content between 0 and 1, both being understood as minimum and maximum effectiveness.

User	Linear Nucleus	Polynomial Nucleus	Radial Nucleus	Sigmoid Nucleus
User 1	0.935	0.903	0.805	0.098
User 2	0.983	0.950	0.400	0.067
User 3	0.933	0.883	0.567	0.400

Table 4.3: Results for 3-fold in the 4 kernels and for each user.

In view of the results obtained by k-fold, the experiments are very satisfactory for the linear kernel. We will then choose this winner because he is also more computationally efficient than the rest.

When testing it in real time, once the model has been trained with the entire data set available, you can see how the SVM machine perfectly meets the requirements of classification. Finally, add that one of the causes of this success also lies in the previous training by the user, that is, we are not only configuring the model to respond to a certain set of data, but we are also educating the body and the person to maintain a certain nature of movements that allows maintain the success of the system.

4.5. Conclusions

During this chapter we wanted to emphasize the interaction of applications with real users. The student realized that the most difficult thing in terms of success in the tests It is not the quality of the software or hardware, but rather it is necessary to have the knowledge to know how to conduct the sessions. React appropriately to the details that make the difference during the experiments it was something learned based on experience in carrying out tests on himself. This is something that is not found in any book or source of information, but, at the same time, it is very important.

Finally, it was possible to verify that both the configuration with the Olimex plate or the configuration with the Aliexpress plate can achieve very good results. Regarding results and physiological differences, we did see that for ectomorph type users the sensor is capable of better recognizing muscle signals. Threshold values are better distinguished and there is more space between movement classes. For some mesomorphs or endomorphs, the power of the read contraction decreased. In all cases it did not represent a problem due to the ability of this plate to adjust its sensitivity.

It would be interesting to try with more age ranges to see if there is a difference. Likewise, we faithfully believe that the methods could be applied to any user with small corrections.

Chapter 5

Development methodology, planning and costs

In this chapter we will detail the concepts applied to the methodology followed for the development of the applications we talked about during this TFG and how it was adjusted to its reality. In addition, a list of costs and time dedicated to the entire process will be provided.

5.1. Methodology

This project was based on the rapid prototyping methodology. It was difficult for us to use any other method where a series of iterations or objectives were defined, since we depended completely on the hardware tools to which we had access and the needs that arose throughout development. Therefore, the software tools created evolved around a changing ecosystem that was taking shape and defining itself throughout the progress of the work.

Likewise, we can make a summary of the phases that were followed and the time spent to each one.

- Phase 1: Acquisition and testing of different hardware devices.
- Phase 2: Assembly of the configuration and checking its correct operation by measuring the first signals.
- Phase 3: Creation of the dataset that will later be analyzed with the software.
- Phase 4: Creation of fully OpenSource software.
- Phase 5: Documentation of the entire process.

- Phase 6: Dissemination of the entire process so that it is useful for anyone in the future interested in the topic.

5.1.1. Phase 1 and 2

Phases 1 and 2 were the ones that took the longest due to the difficulty that currently exists in obtaining information on EMG, on the acquisition of these signals and finally for the purchase of the equipment. Each board has certain characteristics that involved hours of analysis and searching for documentation that would allow us to understand how they work and how they should be used for later application in our development. For the study of the OpenBCI board, the previous experience that the directors of the TFG had was very useful.

5.1.2. Phase 3 and 4

Once our measurement equipment was configured, it was necessary to create software for them and manipulate the datasets. It is for them that these two phases were carried out in parallel. Work was done on the software tools as needs arose and the lines of conduct that offered the best results during the experiments were consolidated. First, it began without any type of graphical interface, creating the first console analysis and measurement programs. Once the threshold and SVM techniques were tested and developed, the idea arose to put everything together in a graphical interface. In this phase, the comments made by an occupational therapist about the needs of real users were very useful.

Likewise, during these phases we could also include the time dedicated to system testing on users, since part of the software was created and tested.

5.1.3. Phase 5 and 6

The last two phases cover the end of this TFG, in which, once learned from all this experience, the student poured all the knowledge gained during these weeks of work and experimentation. He prioritized those aspects that could be useful to anyone interested in the project or that he would have liked to know when he started with this.
The student wrote the report that was reviewed daily by the directors of the TFG.

5.2. Planning

Figure 5.1 shows the planning carried out for the development of the described tasks. previously.

Machine Translated by Google

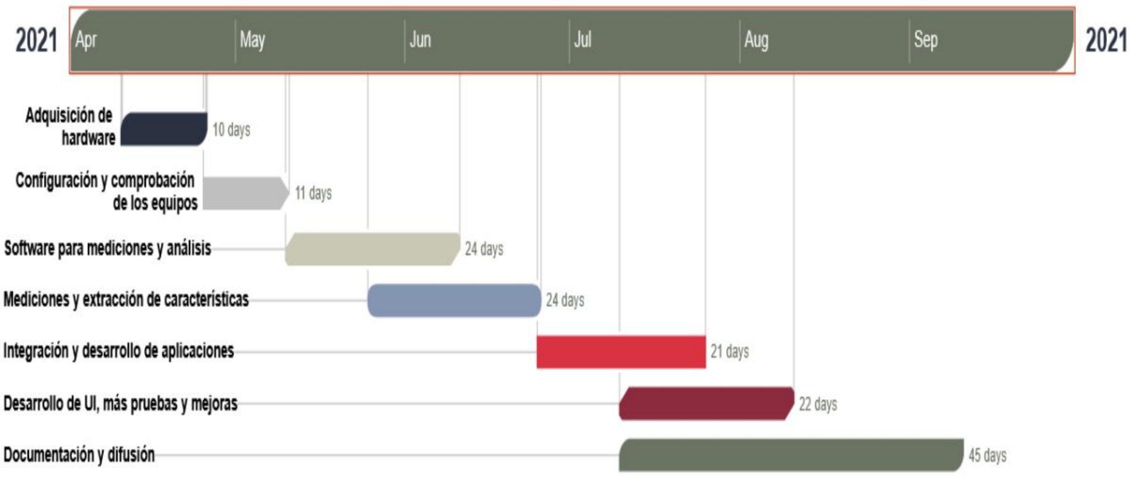


Figure 5.1: Gantt chart of the initial project planning.

The planning that was initially planned was made up of 5 months, starting in the month of April and ending in the month of September 2021. This estimate would be then around 5 hours a day for 5 days a week, which also results in around 25 hours/person per week. If we multiply by the 20 weeks of work, It implies that a total of 500 person hours had been planned.

Phases	Start Date	End Date	Planned Hours
1 and	04/10/21	05/10/21	100
2 3 and	05/10/21	10/8/21	240
4 5 and 6	07/10/21	10/9/21	160

Table 5.1: Initial planning for the project.

In Table 5.1, there is a summary of the initial planning that we had planned. for this Proyect.

5.3. Incidents

Throughout the project we have tried to follow the phase planning as strictly as possible, but due to unforeseen situations and the need for further testing with users, delays arose in the realization and completion of the project.

The most important incidents were:

- Due to the aforementioned difficulty in acquiring hardware, purchasing and shipping It took longer than expected. Even once received, it was necessary to dedicate resources to study its use. For the Aliexpress plates, the seller did not indicate anywhere what their correct configuration was and therefore it was necessary, Spend time and testing until we find the information needed to use it. More of the same happened with Olimex. In the introductory chapters we mentioned that to achieve correct operation a library had to be adapted of filtering in C++ to this board in question that achieved the best results. Due to this incident, phases 1, 2 and 3 were delayed by 60 days compared to planning. initial.

- The first measurements carried out with the equipment did not offer the success we were looking for. All this because of not knowing and not having a clear procedure when it comes to take the samples. We mentioned this same thing during chapter 4, the position of the arm during the sessions, the placement of the electrodes, the correct functioning software and hardware. In an environment where there were so many factors that could being failing at the same time, time and resources were consumed in setting the guidelines that would allow us to take advantage of the material. Due to this incident, phases 3 and 4 were 30 days delayed from initial planning.

- For the correct functioning of the application in the 3D model, it was necessary to dedicate time to understand the existing integration between Blender and Python, while devising a mechanism to connect the EMG readings with the program. It's not about something complex. However, it took a long time to understand how it worked. this integration. Due to this incident, phases 3 and 4 were delayed by about 15 days on initial planning.

- As indicated, the tests were carried out with users outside this project. because the objective is to test it with an amputee patient. It has not been easy to get such a user, but through the COGAMI association, we hope to make these tests soon. Due to what was stated above and combining the accumulated delays, phases 5 and 6 suffered a delay of 60 days.

Phases	Start Date	End Date	Actual Hours
1 and	04/10/21	07/10/21	140
2 3 and	05/10/21	10/10/21	260
4 5 and 6	09/01/21	11/10/21	160

Table 5.2: Temporal summary of the project.

In Table 5.2 you can see a temporal breakdown of what the project would ultimately take. As you can see, phases 3 and 4 are superimposed on the others due to the need to develop the tools as the material arrived, was tested, and emerged.

new needs and the first prototypes were developed. Finally, the project was developed for a total of 560 hours.

5.4. Cost estimation

The cost estimation in this project focuses on the hardware part and human costs. All software used is Open Source and, therefore, is not part of a real cost. The Table 5.3 presents the list of elements that were necessary for the project. Not included the OpenBCI board because it was only used for initial testing and was discarded.

Hardware	Price
Lenovo 530S-14IKB laptop	€58
Arduino Uno	23
Olimex Shield-ekg-emg x2	€48.4
Olimex dry electrodes	€24
Aliexpress PCB + Electrodes x2	€30
Total	€183.4

Table 5.3: Summary of project costs.

For the cost of the laptop, a useful life of 5 years and a use of 7 months has been considered. this project. The rest of the material is fully charged to the project.

Finally, to calculate the cost of human resources, we differentiate between 2 figures: developer and project managers. For the first, a cost was considered at the rate of €20/hour and for the second €30/hour. For the 560 hours of work we calculate what is shown in table 5.4.

Position	Cost	Hours	Cost
Developer	€20/hour	560	€11,200
Project manager	€30/hour	100	€3,000
Total	-	-	€14,200

Table 5.4: Human resources costs.

The developer dedicated all available time to the different phases where he received support and supervision in all tasks by directors.

Finally, adding all the costs mentioned so far, the project has an estimated cost of €14,383.4.

Chapter 6

Conclusions and future lines

In this last chapter, we want to analyze what has been achieved so far in the project and the concerns that have arisen for the student in the process to, finally, talk about the future that eHand intends to pursue.

6.1. Conclusions

This project was born with the purpose of building a cheap and accessible hand prosthesis for anyone. It seems that during this work little is said about this, but in reality the objective evolved to adapt to the real problem that exists around the construction of a system of this type. This problem lies in the fact that the technology is not within the reach of the average individual. It is reserved for a niche that does not represent the majority of the population. Nor is it an accessible technology for people with classical knowledge in the area of computing or electronics. Therefore, with this project we have managed to review good hardware alternatives at a low price and software tools have been created to help this cause. We have proposed different uses that could be given to EMG and we firmly believe that this is the way to follow. It has been shown during experiments that it is easy to achieve good results even with people who have never had contact with a system of this style.

The achievements achieved during this TFG are summarized in:

- We have reviewed and used low-cost material for all experiments.
- A software platform has been created in which to work with EMG signals comfortably and quickly. Furthermore, said platform and all the code developed for the analysis of EMG signals have been published in a public access repository [16], so that any interested person can use it.
- It has been possible to characterize hand movements with both 1 and 2 sensors.

- We have managed to use these features to control human-machine interfaces.
- A good purpose has been given to this very thing. Both for the game and for the control of the 3D prosthesis.
- The documentation carried out includes the most important details and information necessary to help the development of EMG technology.

Not everything went as we expected during the completion of this work, so we must also do an exhaustive analysis of its completion to extract lessons and experiences that will help us for the future. Among the lessons obtained we can highlight:

- Unforeseen events: it is normal for unforeseen events to arise during a development, but we have learned that it is necessary to try to cure them in time. The flexibility of a project is necessary to avoid delays that could be compensated by performing other tasks instead.
- Planning should be an essential part of every project. It is also a useful tool to avoid problems with the previous point. During this project there was a lot of room for improvement in this aspect. However, the nature of a job like this needed time and maturation to take the direction it currently has.
- Finally, lack of experience played an important role during the completion of the work. During development and experimentation, one could have been much more methodical to avoid confusing results or the repetition of experiments due to lack of rigor in those that had already been carried out. In addition, certain practices such as programming first and then designing could have been avoided. There may be advantages to a rapid prototyping methodology, but it is no reason to be careless. Finally, he also highlighted the lack of experience when preparing technical documentation such as this present document.

6.2. Future lines

We believe that this project is not finished at all and, therefore, during this section We will talk about the path we will follow for its development.

The tests carried out were with people who have the hand because we did not have the collaboration of users without this member. Through the COGAMI association, we think that we will soon carry out tests with a real patient. However, this requires a prior study on various aspects such as, for example, type of amputation, muscle strength in the area, etc.

Regarding the application that was taught during this work, the idea would be for it to evolve into an Open Source platform where anyone interested can develop any type of tool. Not only for medical purposes. Its use is contemplated for virtual reality or any other type of human-machine interface controlled by muscle signals.

It would be interesting if the application were able to:

- Allow the user to connect to the Arduino or microcontroller they use wirelessly. We have thought of using the bluetooh connection for this purpose.
- Currently the application is only capable of processing up to 2 channels of EMG signals. We want to increase this number to 4 or 8 in the future.
- The program responsible for training and using the SVM machine is currently not integrated into the application. This will also be resolved.
- We mentioned in previous chapters that the trick to controlling the video game application was to map one of the movements as if it were the “click” of a mouse. Well, in subsequent iterations we want to allow the user to choose which button or action they want to execute with the movement characterization. This will allow you to test the system in different areas than just the jumping game we currently have.
- Give more analysis possibilities, not just RMS. It would be interesting if the application was capable of playing with some more statistical methods. Even the option of being able to apply frequency transformations or filters would also help future users during their research.
- Likewise, it would then be understandable to do an analysis on hardware and software co-design to find a balance that benefits us all. For example, in this TFG we will study support vector models, but these are not the only ones. They are not the only ones, but they do meet the restriction of simplicity and speed in decision making. It could also happen that another classification algorithm such as Neuron Networks or Bayes model combines better with the architecture. In addition to this, there are many other qualities of the hand that could be mimicked and that perhaps involve a series of software mechanisms that combine better with one hardware or another.

  For example, during the project we were able to classify hand movements, but how could we characterize the velocity or angular momentum and even the force at which the movement should be produced based on EMG signals?
- Finally, we want to make corrections to the graphical interface that exists right now, make it more attractive to users and improve the existing software architecture.

behind.

This would be what we have in mind for software tools. What will happen to the hardware? In principle we want to do 2 things: build an electromyograph with Aliexpress PCBs and test some of the boards described during chapter 3. For the first purpose we will use the following tools:

- We want the electromyograph to have a Bluetooth connection. And that is why we plan to use the ESP32 microcontroller as a central element. As can be seen in Figure 6.1, it also has 2 ADC converters and can be powered with an input between +3.3 V and +5 V. There is also the alternative of using an Arduino Nano connected to a Bluetooth expansion board. For now we are opting for the first option that contains everything in the same pack.

- To power the plates we will have a rechargeable battery. You would need a circuit for the load and circuitry to regulate the output voltage. All of these components are cheap and easy to obtain.

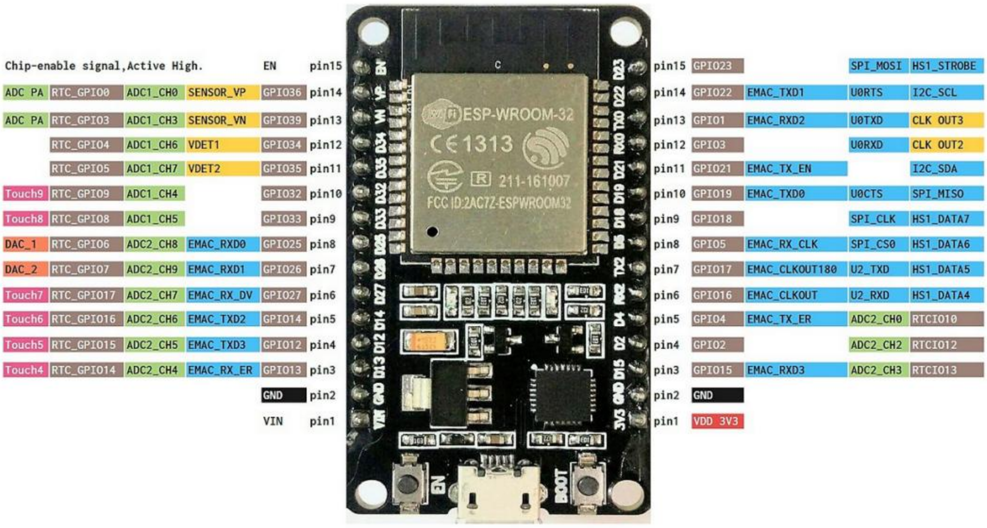


Figure 6.1: ESP32 microcontroller, with Wifi and Bluetooth technology.

Regarding the second objective (continue testing sensors) we plan to acquire the Myoware board for the qualities it offers in design, signal collection and configurability. On the other hand, we will buy new electrodes to try to adapt them to use with the OpenBCI board available at the University, it will be interesting to see the scope of the analyzes that are achieved thanks to the experience with this PCB. In relation to the plates

Olimex, we think that the current filtering library used on it does not get the full benefit we were looking for and therefore, it would be interesting to take a look at the filters it uses to improve its results.

Finally, and although there is much room for improvement in the developed applications, the existing product already allows us to start thinking about the construction of a prototype for a hand prosthesis. The previous manufacturing step for the electromyograph will help us understand the needs that exist when handling certain circuitry and that also exist in 3D modeling and printing.

Bibliography

[1] “Open bionics,” https://openbionics.com/en/hero-arm/, [accessed 10-November-2021].

[2] G. Arnold, L. Ziaoou, G. Fortino, R. Perez, V. Lumelsky, L. Shafer, and D. Goldgof, Surface Electromyography: Physiology, Engineering and Applications. John Wiley & Sons Inc., Hoboken, New Jersey, 2016.

[3] MDURANCE, “What is surface electromyography?” https://blog.mdurance.eu/academia/que-es-la-electromiografia-de-superficie/, 2019.

[4] A. Phinyomark, C. Limsakul, and P. Phukpattaranont, “Application of wavelet analysis in emg feature extraction for pattern classification,” 2011.

[5] “Olimex,” https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG/ open-source-hardware, [accessed 03-September-2021].

[6] “Aliexpress,” https://es.aliexpress.com/, [accessed 10-September-2021].

[7] “Advancer technologies, myoware,” http://www.advancertechnologies.com/p/myoware.html, [accessed 5-November-2021].

[8] “Oymotion, gravity,” https://www.dfrobot.com/product-1661.html, [accessed 5-November-2021].

[9] “Openbci,” https://shop.openbci.com/products/cyton-biosensing-board-8-channel?variant=38958638542, [accessed 20-September-2021].

[10] “Mdurance,” https://mdurance.eu/, [accessed 16-September-2021].

[11] J. Ribeiro, F. Mota, T. Cavalcante, I. Nogueira, V. Gondim, V. Albuquerque, and A. Ale xandria, “Analysis of man-machine interfaces in upper-limb prosthesis: A review,” Robotics, vol. 8, no. 1 p. 16, 2019.

[12] JR Cram, CRAM'S INTRODUCTION TO: Surface Electromyography, 2nd ed. Jones and Bartlett, 1998.

[13] "Ossur, prosthetic solutions," https://www.ossur.com/es-es, [accessed 25-October 2021].

[14] "Biosignalsplux," https://biosignalsplux.com/products/sensors/electromyography.html, [accessed 15-October-2021].

[15] "Canal de biomakers inc." https://www.youtube.com/c/BioMakers, [accessed 25- October-2021].

[16] "ehand repository," https://github.com/aleir97/eHand, [accessed 10-November-2021].

[17] L. van Dijk, C. van der Sluis, H. van Dijk, and R. Bongers, Learning an EMG Controlled Game: Task-Specific Adaptations and Transfer. PLoS One., 2016.

[18] "Official website of biomakers inc." https://biomakers.tech/, [accessed 15-October-2021].

[19] "Altbionics," https://altbionics.com/, [accessed 15-October-2021].

[20] "Youbionc," https://www.youbionic.com/, [accessed 15-October-2021].

[21] "Cogami Association," https://www.cogami.gal/es_ES/, [accessed 5-October-2021].

[22] "Prostheses michelangelo," https://www.ottobock.es/protesica/miembro-superior/sistemas-de-brazo-y-mano/axon-bus-con-mano-michelangelo/, [accessed 16- September-2021].

[23] Arduino, "Documentation of the Arduino Uno R3," https://www.arduino.cc/, [accessed 25-October-2021].

[24] Python, "Documentation for Python," https://docs.python.org/, [accessed 25-October 2021].

[25] Matplotlib, "Main webpage of the Matplotlib python library," https://matplotlib.org/, [ac ceased 25-October-2021].

[26] Numpy, "Main webpage of the Numpy python library," https://numpy.org/about/, [ac ceased 25-October-2021].

[27] PySerial, "Main webpage of the PySerial python library," https://pyserial.readthedocs.io/ en/latest/ index.html#, [accessed 25-October-2021].

[28] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Pret tenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Va roquaux, “API design for machine learning software: experiences from the scikit learn project,” in ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.

[29] “Blender,” https://www.blender.org, [accessed 25-October-2021].

[30] L. Myers, M. Lowery, M. O'Malley, C. Vaughan, C. Heneghan, A. St Clair Gibson, Y. Harley, and R. Sreenivasan, “Rectification and non-linear pre-processing of emg signals for cortico-muscular analysis,” Journal of Neuroscience Methods, vol. 124, no. 2, pp. 157–165, 2003. [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S0165027003000049

[31] AM Sanz and LM Lorente, Advanced EMG Signal Processing. Carlos III University of Madrid, 2017.

[32] Scikit-learn, “Documentation from Scikit-learn,” https://scikit-learn.org/, [accessed 25- October-2021].

[33] “Video game website,” https://www.minijuegos.com/, [accessed 10-November-2021].